(Un)interpretability of Transformers: a case study with Dyck grammars

Anonymous Authors[1]

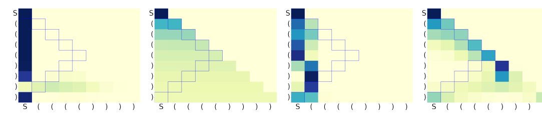
Abstract

Understanding the algorithm implemented by a model is important for trustworthiness when deploying large-scale models, which has been a topic of great interest for interpretability. In this work, we take a critical view of methods that exclusively focus on individual parts of the model, rather than consider the network as a whole. We consider a simple synthetic setup of learning a Dyck language. Theoretically, we show that the set of models that can solve this task satisfies a structural characterization derived from ideas in formal languages (the pumping lemma). We use this characterization to show that the set of optima is qualitatively rich: in particular, the attention pattern of a single layer can be "nearly randomized", while preserving the functionality of the network. We also show via extensive experiments that these constructions are not merely a theoretical artifact: even with severe constraints to the architecture of the model, vastly different solutions can be reached via standard training. Thus, interpretability claims based on individual heads or weight matrices in the Transformer can be misleading.

1. Introduction

Transformer-based models, typically pretrained with next-token prediction objectives, serve as the basis for various applications. Being able to interpret the pretrained solutions is essential for building trustworthiness towards these models. However, certain interpretability methods can be misleading despite being highly intuitive (Jain & Wallace, 2019; Serrano & Smith, 2019; Rogers et al., 2020; Grimsley et al., 2020; Brunner et al., 2020; Meister et al., 2021).

In this work, we aim to understand the theoretical limitation of interpretability methods by characterizing the set of viable solutions. We focus on a particular toy setup in which Transformers are trained to generate *Dyck grammars*, a classic type of formal language grammar consisting of balanced parentheses of multiple types. Dyck is a useful sandbox, as it captures properties like long-range dependency and hierarchical tree-like structure that commonly appear in natural and programming language syntax, and has been an object of interest in many theoretical studies (Hahn, 2020; Yao et al., 2021; Liu et al., 2022b; 2023). Dyck is canonically parsed using a stack-like data structure. Such stack-like patterns (Figure 1) have been observed in the attention heads (Ebrahimi et al., 2020; Yao et al., 2021).



(a) (b) (c) (d)

Figure 1: **Second-layer attention patterns of two-layer Transformers on Dyck** with (a,b) or without (c,d) position embedding: typical attention patterns do *not* exactly match the intuitively interpretable stack-like pattern in Ebrahimi et al. (2020); Yao et al. (2021). The blue boxes indicate the locations of the last unmatched open brackets, as they would appear in a stack-like pattern. All models reach $\geq 97\%$ accuracy (darker color indicates a higher value).

Recent works (Liu et al., 2023; Li et al., 2023) show via explicit constructions of Transformer weights that Transformers are sufficiently expressive to admit very different solutions that perform equally well on the training distribution. This calls into question:

(Q1) Do empirical solutions match the theoretical constructions given in these representational results (Figure 1)? In particular, are interpretable stack-like pattern in Ebrahimi et al. (2018) the norm or the exception?

(Q2) More broadly, can we understand in a principled manner the fundamental obstructions to reliably "reverse engineering" the algorithm implemented by a Transformer by looking at individual attention patterns?

(Q3) Among models that perform (near-)optimally on the training distribution, even if we cannot fully reverse engineer the algorithm implemented by the learned solutions, can we identify properties that characterize performance beyond the training distribution?

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Our contributions. We provide theoretical evidence that individual components (e.g. attention patterns or weights) of a Transformer should not be expected to be interpretable.

- A **perfect balance** condition (Theorem 3.2) on the attention pattern that is sufficient and necessary for 2-layer Transformers with a *minimal first layer* (Assumption 3.1) to predict optimally on Dyck of *any* length. We then show that this condition permits abundant *non-stack-like* attention patterns that do not necessarily reflect any structure of the task, including *uniform* attentions (Corollary 3.3). We show similar results with a *near-optimal* counterpart for bounded-length Dyck (Theorem C.1).
- **Indistinguishability from a single component** (Theorem 3.4) in the sense that any Transformer can be approximated by pruning a larger random Transformer, proved via a *Lottery Ticket Hypothesis* style argument.

We further accompany these theoretical findings with an extensive set of empirical investigations.

Is standard training biased towards interpretable solutions? While both stack-like and non-stack like patterns can process Dyck theoretically, the inductive biases of the architecture or the optimization process may prefer one solution over the other in practice. In Section **??**, based on a wide range of Dyck distributions and model architecture ablations, we find that Transformers that generalize near-perfectly in-distribution (and reasonably well out-of-distribution) do *not* typically produce stack-like attention patterns, showing that the results reported in prior work (Ebrahimi et al., 2018) should not be expected from standard training.

Do non-interpretable solutions perform well in practice? As a corroboration to our theory, in Section E.2, we empirically verify that we can guide Transformers to learn more balanced attention by regularizing for the balance condition, leading to better generalization.

2. Problem Setup

Dyck languages A Dyck language (Schützenberger, 1963) is generated by a context-free grammar, where the valid strings consist of balanced brackets of different types (for example, "[()]" is valid but "([)]" is not). Dyck_k denote the Dyck language defined on k types of brackets. The alphabet of Dyck_k is denoted as $\{1, 2, \cdots, 2k\} \equiv [2k]$, where for each type $t \in [k]$, tokens $2t-1$ and $2t$ are a pair of corresponding open and closed brackets. Dyck languages can be recognized by a push-down automaton. For a string w and $i \le j \in \mathbb{Z}_+$, we use $w_{i:j}$ to denote the substring of w between position i and position j (both ends included). For a valid prefix $w_{1:i}$, the *grammar depth* of $w_{1:i}$, $\mathrm{depth}(w_{1:i})$ is defined as the depth of the stack after processing $w_{1:i}$: $\mathrm{depth}(w_{1:i}) \triangleq$ #Open Brackets in $w_{1:i}$ − #Closed Brackets in $w_{1:i}$.

We overload the same notation $\mathrm{depth}(w_{1:i})$ to also denote the grammar depth of the bracket at position i. We will use $\tau_{i,d}$ to denote a token of type $i \in [2k]$ placed at grammar depth $d \in \mathbb{N}$. We consider bounded-depth Dyck languages following Yao et al. (2021). Specifically,

$$\mathsf{Dyck}_{k,D} := \{w_{1:n} \in \mathsf{Dyck}_k \mid \max_{i\in[n]} \mathrm{depth}(w_{1:i}) \le D\}$$

is a subset of Dyck_k such that the depth of any prefix of a word is bounded by D. While a bounded grammar depth might seem restrictive, it suffices to capture many practical settings; e.g., the level of recursion occurring in natural languages is typically bounded by a small constant (Karlsson, 2007; Jin et al., 2018). We further define the *length-N prefix set* of $\mathsf{Dyck}_{k,D}$ as $\mathsf{Dyck}_{k,D,N} = \{w_{1:N} \mid \exists n \ge N, w_{N+1:n} \in [2k]^{n-N}, s.t.\ w_{1:n} \in \mathsf{Dyck}_{k,D}\}$. Our theoretical setup uses a fixed data distribution $\mathcal{D}_{q,k,D,N}$. Here q intuitively denotes the probability of seeing an open bracket at the next position. The formal definition is deferred to Appendix D.

Training Objectives. Given a model f_θ parameterized by θ, we train with a *next-token prediction* language modeling objective on a given $\mathcal{D}_{q,k,D,N}$. Here the training loss is defined as $\mathcal{L}_\theta(x) = \mathbb{E}_{w_{1:N}\sim\mathcal{D}_{q,k,D,N}}[\frac{1}{N}\sum_{i=1}^{N} l(f_\theta(w_{1:i-1}), e_{w_i})]$. For our theory analysis, we will use mean squared error as l and for experiments, we will use the cross entropy loss following common practice.

Transformer Architectures. We consider a general formulation of Transformer in this work: the l-th layer is parameterized by $\theta^{(l)} := \{W_Q^{(l)}, W_K^{(l)}, W_V^{(l)}, \mathrm{param}(\mathrm{g}^{(l)})\} \in \Theta$, where $W_K^{(l)}, W_Q^{(l)} \in \mathbb{R}^{m_a\times m}$, and $W_V^{(l)} \in \mathbb{R}^{m\times m}$ are the key, query, and value matrices of the attention module; $\mathrm{param}(\mathrm{g}^{(l)})$ are parameters of a feed-forward network $\mathrm{g}^{(l)}$, consisting of fully connected layers, (optionally) LayerNorms and residual links. Given $X \in \mathbb{R}^{d\times N}$, the matrix of d-dimensional features on a length-N sequence, the l-th layer of a Transformer computes

$$f_l(X;\theta^{(l)}) = \mathrm{g}^{(l)}\Big(\mathrm{LN}\Big(W_V^{(l)} X \mathrm{Attn}(X)\Big) + X\Big), \quad (1)$$

$$\text{with } \mathrm{Attn}(X) = \sigma\Big(\mathcal{C} \cdot \frac{(W_K^{(l)}X)^\top (W_Q^{(l)}X)}{\sqrt{d_a}}\Big),$$

where σ is the column-wise softmax operation defined as $\sigma(A)_{i,j} = \frac{\exp(A_{i,j})}{\sum_{k=1}^{N}\exp(A_{k,j})}$, LN represents column-wise LayerNorm operation defined as $\mathrm{LN}(A)_{1:m,j} = \gamma\frac{\mathcal{P}_\perp A_{1:m,j}}{\|\mathcal{P}_\perp A_{1:m,j}\|_2} + \beta$. $\mathcal{P}_\perp$ denotes the projection orthogonal to the $\mathbf{1}\mathbf{1}^\top$ subspace and allows for a compact way to write the mean subtraction in LayerNorm. $\mathcal{C}$ is the causal mask matrix defined as $\mathcal{C}_{i,j} = \mathbb{1}[i \le j]$. We call $\mathrm{Attn}(X)$ the *Attention Pattern* of the Transformer layer l. We consider single-head attentions in this work, whose simplicity further strengthens the messages in this work.

A L-layer Transformer $\mathcal{T}_L$ consists of L above layers, and a word embedding matrix $W_E \in \mathbb{R}^{d\times 2k}$ and a linear decoding head with weight $W_{\text{Head}} \in \mathbb{R}^{2k\times w}$ and bias $b_{\text{Head}} \in \mathbb{R}^{2k}$. Let $\mathcal{Z} \in \mathbb{R}^{2k\times N}$ denote the one-hot embedding of a length-N sequence, then $\mathcal{T}_L$ computes for $\mathcal{Z}$ as

$$\mathcal{T}_L(\mathcal{Z}) = W_{\text{Head}} f_L(\cdots(f_1(W_E\mathcal{Z})) + b_{\text{Head}}. \quad (2)$$

Further, define the *nonstructural pruning* as:

Definition 2.1 (Nonstructural pruning). The *nonstructural pruning* [1] of a Transformer refers to the type of pruning where some entries of the weight matrices are set to zero, and some LayerNorms are set as the identity.

3. Theoretical Analysis

Many prior works have looked for intuitive interpretations of Transformer solutions by studying the attention patterns of particular heads or some individual components of a Transformer (Clark et al., 2019; Vig & Belinkov, 2019; Dar et al., 2022). However, we show next why this methodology can be insufficient even in simple settings. Namely, in Transformer solutions for Dyck, neither attention patterns nor individual local components are guaranteed to encode structures specific for parsing Dyck. We further argue that the converse is also insufficient: when a Transformer does produce interpretable attention patterns, there could be limitations of such interpretation as well, as discussed in Appendix B. Together, our results provide theoretical evidence that careful analyses (beyond heuristics) are required when studying interpretations from Transformer.

3.1. Interpretability Requires Inspecting More Than Attention Patterns

This section focuses on Transformers with 2 layers, which are sufficient for processing Dyck (Yao et al., 2021). We will show that even under this simplified setting, attention patterns alone are not sufficient for interpretation. In fact, we will further restrict the set of 2-layer Transformers by requiring the first-layer outputs to only depend on information necessary for processing Dyck:

Assumption 3.1 (Minimal First Layer). We consider 2-layer Transformers with a *minimal first layer* f_1: let $\boldsymbol{Z} \in \mathbb{R}^{2k\times N}$ denote the one-hot embeddings of inputs $t_1, \ldots, t_N \in [2k]$, then the j_{th} column of the output $f_1(W^E\boldsymbol{Z})$ only depends on the type and depth of t_j, $\forall j \in [N]$.

The Minimal First Layer is a strong condition, as it requires the first layer output to depend only on the bracket type and depth and eliminate all other information, including positions. There are multiple constructions of a minimal first layer, such as the one in (Yao et al., 2021). When working with a minimal first layer, we will not explicitly reason about its parameterization, but instead work directly with its output. Specifically, $\boldsymbol{e}(\tau_{t,d})$ the output of $\tau_{t,d}$ for $t \in [2k]$, $d \in [D]$.

Perfect Balance Condition We find that the attention patterns alone can be too flexible to be helpful, even for the restricted class of a two-layer Transformer with a minimal first layer (Assumption 3.1) and even on a language as simple as Dyck. In particular, the second-layer attention matrix $(W_K^{(2)})^\top W_Q^{(2)}$ only needs to satisfy one condition:

Theorem 3.2 (Perfect Balance, informal). *Consider a two-layer Transformer $\mathcal{T}$ using a minimal first layer with output embeddings $\{\boldsymbol{e}(\tau_{i,d})\}_{d\in[D],i\in[2k]}$. Let $\theta^{(2)} := \{W_Q^{(2)}, W_K^{(2)}, W_V^{(2)}, \text{param}(\text{g}^{(2)})\}$ denote the second layer weights. Under some assumptions on $\theta^{(2)}$, there exist $\{\boldsymbol{e}(\tau_{i,d})\}$ and $\theta^{(2)}$ that minimize the mean squared error (Eqn. 11) on $\mathsf{Dyck}_{k,D}$ for any length N, if and only if $\forall i, j_1, j_2 \in [k], 0 \le d' \le D, 1 \le d_1 \le d_2 \le D$,*

$$(\boldsymbol{e}(\tau_{2i-1,d'+1}) - \boldsymbol{e}(\tau_{2i,d'}))^\top (W_K^{(2)})^\top W_Q^{(2)} \quad (3)$$
$$(\boldsymbol{e}(\tau_{2j_1,d_1}) - \boldsymbol{e}(\tau_{2j_2,d_2})) = 0. \quad (4)$$

Recall that $\boldsymbol{e}(\tau_{2i-1,d'+1}), \boldsymbol{e}(\tau_{2i,d'})$ denote the first-layer outputs of a matching pair. Equation (3) says that since matching brackets do not affect future predictions, their embeddings should balance out each other. It is important to note that the perfect balance condition does not restrict much on the attention patterns. For example, even the uniform attention satisfies the condition and can solve Dyck:

Corollary 3.3. *There exists a two-layer Transformer with uniform attention and without position embedding that can generate the Dyck language of arbitrary length.*

Uniform attention patterns are hardly reflective of any structure of Dyck, hence Corollary 3.3 proves that attention patterns can be oblivious about the underlying task, violating the "faithfulness" criteria for an interpretation (Jain & Wallace, 2019). We will further show in Appendix B.1 that empirically, seemingly structured attention patterns may not accurately represent the inherent structure of the task.

3.2. Interpretability Requires Inspecting More Than Any Single Weight Matrix

Another line of interpretability works involves inspecting the weight matrices of the model (Li et al., 2016; Dar et al., 2022; Eldan & Li, 2023), some of which are done locally, neglecting the interplay between different parts of the model. Our next result shows from a representational perspective that isolating single weights may also be misleading:

Theorem 3.4 (Indistinguishability From a Single Component, informal). *Consider a L-layer Transformer $\mathcal{T}$*

[1] As opposed to *structural pruning* which prunes some channels of weight matrices.

with embedding dimension m, width w and $g^{(k)}(x) = \mathrm{LN}\left(W_2^{(k)}\mathrm{ReLU}\left(W_1^{(k)}x\right)\right) + x$. Consider a polynomial larger random *Transformer $\mathcal{T}_{large}$, with $4L$ layers, embedding dimension $4m$, and width $O(\max\{m\log\frac{wmLN}{\epsilon\delta}, w\})$, and same architecture choice for g, whose weights are sampled as $W_{i,j} \sim U(-1,1)$ for every $W \in \mathcal{T}_{large}$. Then, with probability $1-\delta$ over the randomness of $\mathcal{T}_{large}$, a nonstructural pruning (Definition 2.1) of $\mathcal{T}_{large}$, denote $\mathcal{T}'_{large}$, can ϵ-approximate $\mathcal{T}$. That is, $\forall \boldsymbol{X} \in \mathbb{R}^{d\times N}$ with $\|\boldsymbol{X}_{:,i}\|_2 \le 1$, $\forall i \in [N]$, $\|\mathcal{T}'_{large}(\boldsymbol{X}) - \mathcal{T}(\boldsymbol{X})\|_2 \le \epsilon$.*

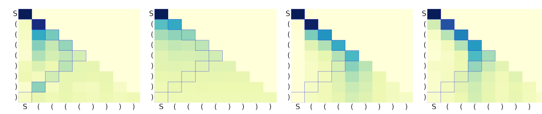
Moreover, pick any $W \in \mathcal{T}_{large}$, with probability $1-\delta$, for any smaller Transformers $\mathcal{T}_1, \mathcal{T}_2$ satisfying same conditions as $\mathcal{T}$, we have two pruned Transformers $\mathcal{T}_{Large,1}, \mathcal{T}_{Large,2}$ based on $\mathcal{T}_{large}$, such that they coincide on the pruned weight of W, and $\mathcal{T}_{Large,i}$ ϵ-approximate $\mathcal{T}_i$, $\forall i \in \{1,2\}$.

4. Experiments: Varoius Dyck Solutions

Our theory in Section 3 proves the existence of abundant *non-stack-like* attention patterns, all of which suffice for (near-)optimal generalization on Dyck. However, could there be *implicit biases* in the architecture and the optimization algorithm, which would potentially make the learned attention patterns more frequently stack-like? In this section, we show there is no evidence for such implicit bias in standard training .We will also show a modified objective based on our theory can be used to *explicitly regularize* the model towards better length generalization (Section E.2).

We empirically verify our theoretical findings that Dyck solutions can give rise to a variety of attention patterns. We use the Adam optimizer (Kingma & Ba, 2014) unless specified otherwise. We use Transformers with 2 layers, 1 head, hidden dimension 50 and word embedding dimension 50. We test the accuracy of the model by randomly generating a Dyck prefix that ends with a closing bracket, and evaluating whether the model predicts correctly the type of the last closing bracket given the rest of the prefix. Note that in this setting a correct parser should always be able to uniquely determine the correct closing bracket type (for the sequence to remain a valid Dyck sequence). We train on valid $\mathrm{Dyck}_{2,4}$ sequence with length less than 28 generated with $q = 0.5$, where all models are able to achieve $\ge 97\%$ test accuracy.

Qualitative Results. As a response to (Q1), we observe that attention patterns of Transformers trained on Dyck are not always stack-like (Figure 1). In fact, the attention patterns vary even across different random initializations. Moreover, while Theorem 3.2 predicts that position encoding is not necessary for a Transformer to generate Dyck (this is verified by experiments, as Transformers with no positional encoding achieve $\ge 97\%$ accuracy), we observe that adding the position encoding [2] does affect the attention patterns. We also try fixing the attention layer as uniform attention and verify that they can also fit the distribution almost perfectly, which is consistent with our theory.



(a) Embedding Type 1 (b) Embedding Type 1 (c) Embedding Type 2 (d) Embedding Type 3

Figure 2: **Second-layer attention patterns of two-layer Transformers with a minimal first layer**: (a), (b) are based on embedding Type 1 with different learning rates, where the attention patterns show much variance as Theorem 3.2 predicts. (c), (d) are based on embedding Type 2 and Type 3. Different embedding functions lead to diverse attention patterns, most of which are not stack-like.

We then experiment with two-layer Transformers with a minimal first layer. We experiment with three different types of embeddings $\boldsymbol{e}$, the exact format is shown in Appendix E.1. As one can observe from Figure 2, the attention patterns learned by Transformers exhibit large variance between different choices of architectures and learning rates. We observe that most of the attention patterns learned by the Transformer are not stack-like.

Quantiative Experiments. We now quantify the variation in attention by comparing across multiple random initializations. We define the *attention variation* between two attention patterns $A_1, A_2 \in \mathbb{R}^{N\times N}$ over an length-N input sequence as $\mathrm{Variation}(A_1, A_2) = \|A_1 - A_2\|_F^2$. We will then calculate the average variation of an architecture by running $n = 40$ random initializations and calculate the average variation between the attention patterns of the n random initializations on sequence [[[[]]]](((()))). We will call this quantity the *average attention variation*.

We observe that for standard two layer training with linear position embedding, the average attention variation is 2.20. For training without position embedding, the average attention variation is 2.27. Both variation is closed to the random baseline value of 2.85 [3], showing that the attention head learned by different initializations indeed tend to be very different. We also experiment with Transformer with a minimal first layer and the embedding in Equation (Type 1), which reduces the average variation to 0.24. We hypothesize that the structural constraints in this setting provide sufficiently strong inductive bias that limit the variability of attention patterns.

[2] We use the linear positional encoding following (Yao et al., 2021), where for the i_{th} position, define encoding $\boldsymbol{e}_p(i) := i/T_{\max}\cdot$ for some $T_{\max}$.

[3] The random baseline is calculated by generating purely random attention patterns (from the simplex, i.e. random square matrices s.t. each row sums up to 1) and calculate the average attention variation between them.

References

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1): 207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.576. URL https://aclanthology.org/2020.emnlp-main.576.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.37. URL https://aclanthology.org/2020.conll-1.37.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. An interpretability illusion for bert. *arXiv preprint arXiv: Arxiv-2104.07143*, 2021.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJg1f6EFDB.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 2020. doi: 10.23915/distill.00024. https://distill.pub/2020/circuits.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space, 2022.

Yichuan Deng, Zhihang Li, and Zhao Song. Attention scheme inspired softmax regression, 2023.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *International Conference on Computational Linguistics (COLING)*, 2018.

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4301–4306, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.384. URL https://aclanthology.org/2020.findings-emnlp.384.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5793–5831. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/edelman22a.html.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://Transformer-circuits.pub/2021/framework/index.html.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression, 2023.

F. Gers and J. Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE transactions on neural networks*, 12 6:1333–40, 2001.

Christopher Grimsley, Elijah Mayfield, and Julia R.S. Bursten. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1780–1790, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.220.

Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Trans. Assoc. Comput. Linguistics*, 8:156–171, 2020. doi: 10.1162/tacl_a_00306. URL https://doi.org/10.1162/tacl_a_00306.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, 2019.

John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D Manning. Rnns can generate bounded hierarchical languages with optimal memory. *arXiv preprint arXiv:2010.07515*, 2020.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. Do attention heads in bert track syntactic dependencies?, 2019.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357.

Samy Jelassi, Michael Eli Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=eMW9AkXaREI.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. Unsupervised grammar induction with depth-bounded pcfg. *Transactions of the Association for Computational Linguistics*, 6:211–224, 2018.

Fred Karlsson. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392, 2007.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4365–4374, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL https://aclanthology.org/D19-1445.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 681–691, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1082. URL https://aclanthology.org/N16-1082.

Xian Li and Hongyu Gong. Robust optimization for multilingual translation with imbalanced data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 25086–25099. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/d324a0cc02881779dcda44a675fdcaaa-Paper.pdf.

Yuchen Li and Andrej Risteski. The limitations of limited context for constituency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2675–2687, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.208. URL https://aclanthology.org/2021.acl-long.208.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding, 2023.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 241–253, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4825. URL https://aclanthology.org/W19-4825.

Bingbin Liu, Daniel Hsu, Pradeep Kumar Ravikumar, and Andrej Risteski. Masked prediction: A parameter identifiability view. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=Hbvlb4D1aFC.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=De4FYqjFueZ.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. *arXiv preprint arXiv:2210.14199*, 2022b.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5747–5763, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.463. URL https://aclanthology.org/2020.emnlp-main.463.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL https://aclanthology.org/N19-1112.

Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. Is sparse attention more interpretable? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 122–129, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.17. URL https://aclanthology.org/2021.acl-short.17.

William Merrill. Sequential neural networks as automata. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pp. 1–13, Florence, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3901. URL https://www.aclweb.org/anthology/W19-3901.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, November 2-3 2019. Association for Computational Linguistics. URL https://aclanthology.org/2019.iwslt-1.17.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://Transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. *Advances in neural information processing systems*, 33:2599–2610, 2020.

Jorge Perez, Pablo Barcelo, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine Learning*

Research, 22(75):1–35, 2021. URL http://jmlr.org/papers/v22/20-302.html.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3208–3229, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.259. URL https://aclanthology.org/2020.emnlp-main.259.

Alessandro Raganato and Jörg Tiedemann. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL https://aclanthology.org/W18-5431.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

M.P. Schützenberger. On context-free languages and push-down automata. *Information and Control*, 6(3):246–264, 1963. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(63)90306-1. URL https://www.sciencedirect.com/science/article/pii/S0019995863903061.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL https://aclanthology.org/P19-1282.

Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pp. 440–449, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130432. URL https://doi.org/10.1145/130385.130432.

Kaiser Sun and Ana Marasović. Effective attention sheds light on interpretability. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4126–4135, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.361. URL https://aclanthology.org/2021.findings-acl.361.

Mirac Suzgun, Yonatan Belinkov, Stuart Shieber, and Sebastian Gehrmann. LSTM networks can perform dynamic counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pp. 44–54, Florence, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3905. URL https://www.aclweb.org/anthology/W19-3905.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 63–76, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4808. URL https://aclanthology.org/W19-4808.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers, 2021. URL https://arxiv.org/abs/2107.13163.

Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv:1805.04908*, 2018.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11080–11090. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/weiss21a.html.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and* the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4166–4176, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.383. URL https://aclanthology.org/2020.acl-main.383.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3770–3785, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.292. URL https://aclanthology.org/2021.acl-long.292.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 2020.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task, 2022. URL https://arxiv.org/abs/2206.04301.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word?, 2023.

Appendix

A. Related Work

There has been a flourishing line of work on interpretability in natural language processing. Multiple "probing" tasks have been designed to extract syntactic or semantic information from the learned representations (Raganato & Tiedemann, 2018; Liu et al., 2019; Hewitt & Manning, 2019; Clark et al., 2019). However, the effectiveness of probing often intricately depend on the architecture choices and task design, and sometimes may even result in misleading conclusions (Jain & Wallace, 2019; Serrano & Smith, 2019; Rogers et al., 2020; Brunner et al., 2020; Prasanna et al., 2020; Meister et al., 2021). While these challenges do not completely invalidate existing approaches (Wiegreffe & Pinter, 2019), it does highlight the need for more fundamental understanding of interpretability.

Towards this, we choose to focus on the synthetic setup of Dyck whose solution space is easier to characterize than natural languages, allowing us to identify a set of feasible solutions. While similar representational results have been studied in prior work (Yao et al., 2021; Liu et al., 2023; Zhao et al., 2023), our work emphasizes that theoretical constructions do not resemble the solutions found in practice. Moreover, the multiplicity of valid constructions suggest that understanding Transformer solutions require analyzing the optimization process, which a number of prior work has made progress on (Jelassi et al., 2022; Li et al., 2023; Deng et al., 2023).

Finally, it is worth noting that the challenges highlighted in our work do not contradict the line of prior works that aim to improve *mechanistic interpretability* into a trained model or the training process (Elhage et al., 2021; Olsson et al., 2022; Nanda et al., 2023; Li et al., 2023), which aim to develop circuit-level understanding of a particular model or the training process.

Interpreting Transformer solutions Prior empirical works show that Transformers trained on natural language data can produce representations that contain rich syntactic and semantic information, by designing a wide range of "probing" tasks (Raganato & Tiedemann, 2018; Liu et al., 2019; Hewitt & Manning, 2019; Clark et al., 2019; Tenney et al., 2019; Hewitt & Liang, 2019; Kovaleva et al., 2019; Lin et al., 2019; Wu et al., 2020; Belinkov, 2022) (or other approaches using the attention weights or parameters in neurons directly Vig & Belinkov, 2019; Htut et al., 2019; Sun & Marasović, 2021; Eldan & Li, 2023). However, there is no canonical way to probe the model, partially due to the huge design space of probing tasks, and even a slight change in the setup may lead to very different (sometimes even seemingly contradictory) interpretations of the result (Hewitt & Liang, 2019). In this work, we tackle such ambiguity through a different perspective—by developing formal (theoretical) understanding of solutions learned by Transformers. Our results imply that it may be challenging to try to interpret Transformer solutions based on individual parameters (Li et al., 2016; Dar et al., 2022), or based on constructive proofs (unless the Transformer is specially trained to be aligned with a certain algorithm, as in Weiss et al., 2021).

Interpreting attention patterns Prior works (Jain & Wallace, 2019; Serrano & Smith, 2019; Rogers et al., 2020; Grimsley et al., 2020; Brunner et al., 2020; Prasanna et al., 2020; Meister et al., 2021; Bolukbasi et al., 2021, *inter alia*) present negative results on deriving explanations from attention weights using approaches by Vig & Belinkov (2019); Kobayashi et al. (2020, *inter alia*). However, Wiegreffe & Pinter (2019) argues to the contrary by pointing out flaws in the experimental design and arguments of some of the prior works; they also call for theoretical analysis on the issue. Hence, a takeaway from these prior works is that expositions on explainability based on attention requires clearly defining the notion of explainability adopted (often task-specific). In our work, we restrict our main theoretical analysis to the fully defined data distribution of Dyck language (Definition D.1), and define "interpretable attention pattern" as the stack-like pattern proposed in prior theoretical (Yao et al., 2021) and empirical (Ebrahimi et al., 2020) works. These concrete settings and definitions allow us to mathematically state our results and provide theoretical reasons.

Theoretical understanding of representability Methodologically, our work joins a long line of prior works that characterize the solution of neural networks via the lens of simple synthetic data, from class results on RNN representability (Siegelmann & Sontag, 1992; Gers & Schmidhuber, 2001; Weiss et al., 2018; Suzgun et al., 2019; Merrill, 2019; Hewitt et al., 2020), to the more recent Transformer results on parity (Hahn, 2020), Dyck (Yao et al., 2021), topic model (Li et al., 2023), and formal grammars in general (Bhattamishra et al., 2020a; Li & Risteski, 2021; Zhang et al., 2022; Liu et al., 2023; Zhao et al., 2023). Our work complements prior works by showing that although representational results can be obtained via intuitive "constructive proofs" that assign values to the weight matrices, the model does not typically converge to those intuitive solutions in practice. Similar messages are conveyed in Liu et al. (2023), which presents different types of constructions using different numbers of layers. In contrast, we show that there exist multiple different constructions even when the number of layers is kept the same.

There are also theoretical results on Transformers in terms of Turing completeness (Bhattamishra et al., 2020b; Perez et al., 2021), universal approximatability (Yun et al., 2020), and statistical sample complexity (Wei et al., 2021; Edelman et al., 2022), which are orthogonal to our work.

Transformer optimization Given multiple global optima, understanding Transformer solutions requires analyzing the training dynamics. Recent works theoretically analyze the learning process of Transformers on simple data distributions, e.g. when the attention weights only depend on the position information (Jelassi et al., 2022), or only depend on the content (Li et al., 2023). Our work studies a syntax-motivated setting in which both content and position are critical. We also highlight that Transformer solutions are very sensitive to detailed changes, such as positional encoding, layer norm, sharpness regularization (Foret et al., 2020), or pre-training task (Liu et al., 2022a). On a related topic but towards different goals, a series of prior works aim to improve the training process of Transformers with algorithmic insights (Nguyen & Salazar, 2019; Xiong et al., 2020; Liu et al., 2020; Zhang et al., 2020; Li & Gong, 2021, *inter alia*). An end-to-end theoretical characterization of the training dynamics remains an open problem; recent works that propose useful techniques towards this goal include Gao et al., 2023; Deng et al., 2023.

Mechanistic interpretability Finally, it is worth noting that the challenges highlighted in our work do not contradict the line of prior works that aim to improve *mechanistic interpretability* into a trained model or the training process (Cammarata et al., 2020; Elhage et al., 2021; Olsson et al., 2022; Nanda et al., 2023; Li et al., 2023): although we prove that components (e.g. attention scores) of trained Transformers do not generally admit intuitive interpretations based on the data distribution, it is still possible to develop circuit-level understanding about a particular model, or measures that closely track the training process, following these prior works.

A.1. Limitations and future work.

Our results do not preclude that interpretable attention patterns can emerge in multi-head, overparameterized Transformers trained on more complex data distributions. In that case, we discuss some limitations of such interpretation in Appendix B.

Interesting directions of future work include extending our theoretical results to more complex settings (in terms of both architecture choice and data distribution), theoretical characterization of the learning dynamics, and more experiments in controlled settings for testing the connections between the training approach, interpretability, and task performance. We motivate these questions and discuss some relevant trade-offs in Appendix B.

B. Are interpretable attention patterns useful?

Our results Section 3 and Section **??** demonstrate that Transformers are sufficiently expressive that a (near-)optimal loss on Dyck languages can be achieved by a variety of attention patterns, many of which may not be interpretable.

However, multiple prior works have shown that for multi-layer multi-head Transformers trained on natural language datasets, it is often possible to locate attention heads that produce interpretable attention patterns (Vig & Belinkov, 2019; Htut et al., 2019; Sun & Marasović, 2021). Hence, it is also illustrative to consider the *"converse question"* of (Q1): when some attention heads do learn to produce attention patterns that suggest intuitive interpretations, what benefits can they bring?

We discuss this through two perspectives:

- **Reliability of interpretation:** Is the Transformer necessarily implementing a solution consistent with such interpretation based on the attention patterns? (Section B.1)
- **Usefulness for task performance:** Are those interpretable attention heads more important for the task than other uninterpretable attention heads? (Section B.2)

We present preliminary analysis on these questions, and motivate future works on the interpretability of attention patterns using rigorous theoretical analysis and carefully designed experiments.

B.1. Can interpretable attention patterns be misleading?

We show through a simple argument that interpretations based on attention patterns can sometimes be misleading, as we formalize in the following proposition:

Proposition B.1. *Consider an L-layer Transformer $\mathcal{T}$ (Equation (2)). For any $W_K^{(l)}, W_Q^{(l)} \in \mathbb{R}^{m_a \times m}$ $(l \in [L])$, there exist $W_{\text{Head}} \in \mathbb{R}^{2k \times w}$ and $b_{\text{Head}} \in \mathbb{R}^{2k}$ such that $\mathcal{T}(\mathcal{Z}) = 0, \forall \mathcal{Z}$.*

While its proof is trivial (simply setting $W_{\text{Head}} = 0$ and $b_{\text{Head}} = 0$ suffices), Proposition B.1 implies that the solution represented by the Transformer could possibly be independent of the attention patterns in all the layers (1 through l). Hence, it could be misleading to interpret Transformer solutions solely based on these attention patterns.

Empirically, Transformers trained on Dyck indeed sometimes produce misleading attention patterns.

We present one representative example in Figure 3, and Figure 4, in which *all interpretable attention patterns are misleading*.

We also present additional results in Figure 5, in which *some interpretable attention patterns are misleading, and some are not*.

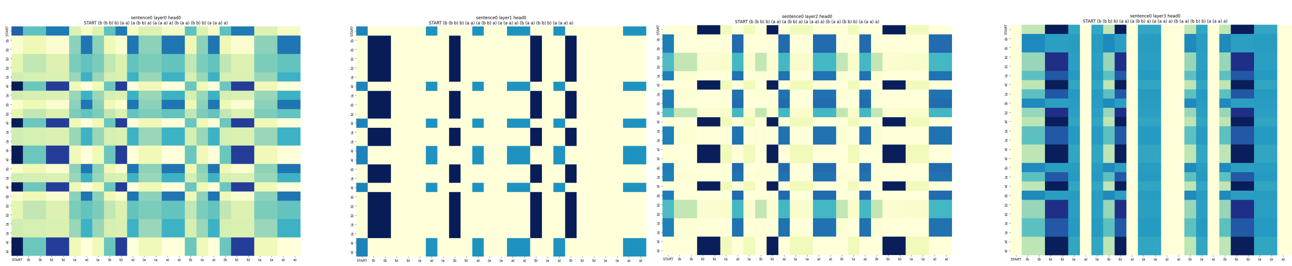


Figure 3: **Even interpretable attention patterns can be misleading**: For a 4-layer Transformer trained on Dyck with the *copying* task (with $> 96\%$ validation accuracy), i.e. the output should be exactly the same as the input, the attention patterns in some layers seem interpretable: (layer 2) attending to bracket type a) or (b; (layer 3) attending to closing bracketss; (layer 4) neve attending to bracket type a); However, none of them are informative of the copying task. This is possible because Transformers can use the residual connections (or weights MLPs or the value matrices) to solve copying, bypassing the need of using attention.

Similar message has been conveyed in prior works (Bolukbasi et al., 2021), and future works may aim to achieve the *faithfulness*, *completeness*, and *minimality* conditions in (Wang et al., 2023).

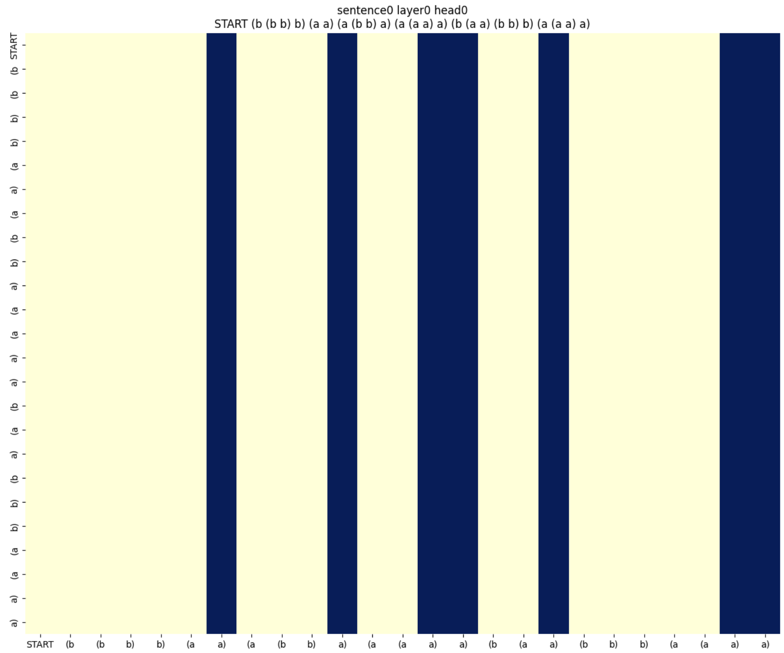
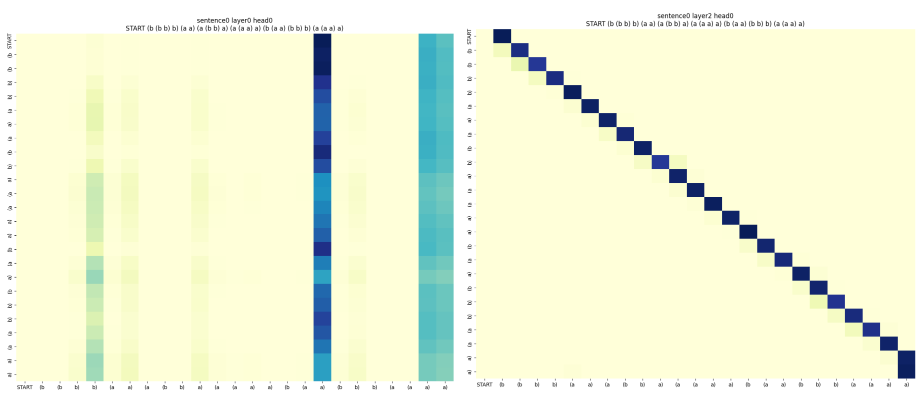


Figure 4: **Even interpretable attention patterns can be misleading**: For a 1-layer Transformer trained on Dyck with the *copying* task (with $> 90\%$ validation accuracy), i.e. the output should be exactly the same as the input, the attention pattern seems to be attending to closing brackets only, but that is not informative of the copying task.



(a) layer 1 of 4

(b) layer 3 of 4

Figure 5: **Even interpretable attention patterns can be misleading**: For a 4-layer Transformer trained on Dyck with the *copying* task (with $> 96\%$ validation accuracy), i.e. the output should be exactly the same as the input, both types of attention patterns are common: (a) attending to closing bracketss, which is uninformative of the copying task; (b) attending to the current position, which solves the copying task.

B.2. Can interpretable attention patterns be important?

Kovaleva et al. (2019) observes that, when the "importance" of an attention head is defined as the performance drop the model suffers when the head is disabled, then for most tasks they test, the most important attention head in each layer *does not* tend to be interpretable.

However, experiments by Voita et al. (2019) led to a seemingly contradictory observation: when attention heads are systematically pruned by finetuning the Transformer with a relaxation of L_0-penalty (i.e. encouraging the number of remaining attention heads to be small), most remaining attention heads that survive the pruning can be associated with certain functionalities such as positional, syntactic, or attending to rare tokens.

These works seem to bring mixed conclusions to our question: are interpretable attention heads more important for the task than other uninterpretable attention heads? We interpret these results by conjecturing that the definition of "importance" (reflected in their experimental design) plays a crucial role:

- When the importance of an attention head is defined *treating all other attention heads as fixed*, motivating experiments that prune/disable certain heads while keeping other heads unchanged (Michel et al., 2019; Kovaleva et al., 2019), the conclusion may be mostly pessimistic: mostly no strong connection between interpretability and importance.
- On the other hand, when the importance of an attention head is defined *allowing all other attention heads to adapt to its change*, motivating experiments that jointly optimize all attention heads while penalizing the number of heads (Voita et al., 2019), the conclusion may be more optimistic: the heads obtained as a result of this optimization tend to be interpretable.

We think the following trade-offs apply:

- On one hand, the latter setting is more practical, since Transformers are typically not trained to explicitly ensure that the model performs well when a single attention head is individually disabled; rather, it would be more intuitive to think of a group of attention heads as jointly representing some transformation, so when one head is disabled, other heads should be fine-tuned to adapt to the change.
- On the other hand, when all other heads change too much during such fine-tuning, the resulting set of attention heads no longer admit an unambiguous one-to-one map with the original set of (unpruned) attention heads. As a result, the interpretability and importance obtained from the set of pruned heads do not necessarily imply those properties of the original heads.

A comprehensive study of this question involves multi-head extensions of our theoretical results (Section 3), and carefully-designed experiments that take the above-mentioned trade-offs into consideration. We think these directions are interesting future work.

C. Approximate Balance Condition For Finite Length Training Data

The condition in Theorem 3.2 requires the model to reach the optimal loss for data of any length. However, in practice, one can only train the model on *finite-length* data and the model can only reach a low but non-optimal loss for finite length data. In this case, the condition in Theorem 3.2 is not precisely met. However, one can show that a similar condition is still necessary if one restricted the Lipschitz constant of the projection function g. We first define two quantities that measure the deviation from the previous ideal scenario:

$$S_{d,d',i,j}[\theta^{(2)}] = \left\| u(\tau_{2j,d}, \tau_{2i,d'}) + u(\tau_{2j,d}, \tau_{2i-1,d'+1}) \right\|_2, \tag{5}$$

$$\boldsymbol{t} = \arg\min_{\boldsymbol{t}\in[k]^d} \Big\| \sum_{d'\le d} u(\tau_{2j,d}, \tau_{2\boldsymbol{t}_{d'},d'}) + u(\tau_{2j,d}, \tau_{2j-1,d+1}) + u(\tau_{2j,d}, \tau_{2j,d}) \Big\|_2. \tag{6}$$

$$P_{d,j}[\theta^{(2)}] = \min_{\boldsymbol{t}'\in[k]^d, \boldsymbol{t}'_d\neq \boldsymbol{t}_d} \Big\| \sum_{d'\le d} u(\tau_{2j,d}, \tau_{2\boldsymbol{t}_{d'},d'}) + u(\tau_{2j,d}, \tau_{2j-1,d+1}) + u(\tau_{2j,d}, \tau_{2j,d}) \Big\|_2. \tag{7}$$

The first term $S_{d,d',i,j}[\theta^{(2)}]$ measures the change in the input of the LayerNorm layer for the last token $\tau_{2j,d}$, when a matching pair of brackets $(\tau_{2i,d'}, \tau_{2i-1,d'+1})$ is inserted into the prefix. Under the perfect balance condition, $S_{d,d',i,j}[\theta^{(2)}] = 0$. The second term $P_{d,j}[\theta^{(2)}]$ is measures the norm of the input of the LayerNorm layer at last token $\tau_{2j,d}$. when the prefix only contains open brackets. In the following theorem, $P_{d,j}$ will be used as a baseline to show $S_{d,d',i,j}[\theta^{(2)}]$ cannot be too large, i.e., the model should not be sensitive to the insertion of a matching pair of brackets.

Theorem C.1 (Approximate Balance)**.** *Consider a two-layer Transformer $\mathcal{T}$ with a minimal first layer trained with the mean squared error (Equation (11)). For any $\gamma, N > 0$ and sufficiently small ϵ, suppose $\mathrm{g}^{(2)}$ is γ-Lipschitz, and suppose the set of second-layer weights $\bar{\theta}^{(2)}_N$ satisfies that $\mathcal{L}(\mathcal{T}[\bar{\theta}^{(2)}_N], \mathcal{D}_{q,k,D,N}) \le q^{-N}\epsilon$. Then, there exists a constant $C_{\gamma,\epsilon,D}$, such that for any $0 \le d' \le D, 1 \le d \le D, i,j \in [k]$, it holds that*

$$S_{d,d',i,j}[\bar{\theta}^{(2)}_N] \le \frac{C_{\gamma,\epsilon,D}}{N} P_{d,j}[\bar{\theta}^{(2)}_N]. \tag{8}$$

Equation (8) requires $S_{d,d',i,j}[\theta^{(2)}]$ to be small relative to $P_{d,j}[\bar{\theta}^{(2)}_N]$, and can be interpreted as a relaxation of **??** which is equivalent to $S_{d,d',i,j}[\theta^{(2)}] = 0$. The proof of Theorem C.1 shares similar intuition as Theorem 3.2 and is given in Appendix D.3. As a direct corollary of Theorem C.1, we can additionally consider adding a weight decay, in which case approximate balance condition holds as the regularization strength goes to 0:

Corollary C.2. *Consider the setting where a Transformer with a fixed minimal first layer is trained to minimize $\mathcal{L}^{reg}_\lambda = \mathcal{L}_\theta(x) + \lambda\frac{\|\theta\|_2^2}{2}$, which is the squared loss with λ weight decay. Suppose the function $\mathrm{g}^{(2)}$ of the Transformer is a fully connected network. Then, for any length N, there exists constant $C > 0$, such that for parameters $\theta_{\lambda,N}$ minimizing $\mathcal{L}^{reg}_\lambda$, it holds $\forall 0 \le d' \le D, 1 \le d \le D, i,j \in [k]$ that,*

$$\limsup_{\lambda\to 0} \frac{S_{d,d',i,j}[\theta_{\lambda,N}]}{P_{d,i}[\theta_{\lambda,N}] + 1} \le \frac{C}{N}.$$

D. Omitted Proofs in Section 3

D.1. Detailed Setup

Data Distribution We will first formaly define the distribution we are considering.

Definition D.1 (Dyck distribution)**.** The distribution $\mathcal{D}_{q,k,D,N}$, specified by $q \in (0,1)$, is defined over $\mathsf{Dyck}_{k,D,N}$ such that $\forall w_{1:N} \in \mathsf{Dyck}_{k,D,N}$,

$$\mathbb{P}(w_{1:N}) \propto (q/k)^{\#\{i|w_i \text{ is open, } \mathrm{depth}(w_{1:i})>1\}} \times (1-q)^{\#\{i|w_i \text{ is closed, } \mathrm{depth}(w_{1:i})<D-1\}}. \tag{9}$$

That is, $q \in (0,1)$ denote the probability of seeing an open bracket at the next position, except for two corner cases: 1) the next bracket has to be open if the current grammar depth is 0 (1 after seeing the open bracket); 2) the next bracket has to be closed if the current grammar depth is D.

Loss Function the next token prediction task uses f_θ to predict the next token for any fixed prefix. Precisely, given a prefix $w_{1:N} \in \mathsf{Dyck}_{k,D,N}$ and a loss function $l(\cdot,\cdot) \to \mathbb{R}$, f_θ is trained to minimize the loss function $\min_\theta \mathcal{L}_\theta(x)$ for

$$\mathcal{L}_\theta(x) = \mathbb{E}_{w_{1:N}\sim\mathcal{D}_{q,k,D,N}}[\frac{1}{N}\sum_{i=1}^{N} l(f_\theta(w_{1:i-1}), e_{w_i})]. \tag{10}$$

We will also consider a ℓ_2-regularized version $\mathcal{L}_\theta^{\text{reg}}(x) = \mathcal{L}_\theta(x) + \lambda\frac{\|\theta\|_2^2}{2}$ with parameter $\lambda > 0$.

For our theory, we will consider the mean squared error (MSE) as the loss function,

$$l := l_{sq}(x, e_i) = \|x - e_i\|_2^2. \tag{11}$$

In our experiments, we apply the cross entropy loss following common practice.

D.2. Proof of Theorem 3.2

The key step is already shown in Section 3. We will restate the proof rigorously here.

Theorem D.2 (Perfect Balance; formal version of Theorem 3.2)**.** *Consider a two-layer Transformer $\mathcal{T}$ with a minimal first layer with output embeddings $\{\boldsymbol{e}(\tau_{i,d})\}_{d\in[D],i\in[2k]}$. Let $\theta^{(2)} := \{W_Q^{(2)}, W_K^{(2)}, W_V^{(2)}, \mathrm{param}(\mathrm{g}^{(2)})\}$ denote the second layer weights.*

Define the balance condition *to be the condition that for any $i, j_1, j_2 \in [k]$ and $d', d_1, d_2 \in [D]$,*

$$(\boldsymbol{e}(\tau_{2i-1,d'}) - \boldsymbol{e}(\tau_{2i,d'-1}))^\top (W_K^{(2)})^\top W_Q^{(2)} (\boldsymbol{e}(\tau_{2j_1,d_1}) - \boldsymbol{e}(\tau_{2j_2,d_2})) = 0. \tag{12}$$

Then, for the existence of $\{\boldsymbol{e}(\tau_{i,d})\}$ and $\theta^{(2)}$ that achieves the Bayes-optimal loss for the mean squared error (Eqn. 11) on $\mathsf{Dyck}_{k,D}$ for any length N, it holds that:

- *If $W_V^{(2)}$ satisfies $\mathcal{P}_\perp W_V^{(2)}\boldsymbol{e}(\tau_{t,d}) \neq 0, \forall t \in [k], d \in [D]$ then the balanced condition is necessary to show existence.*
- *Conversely, if the set of $2k$ encodings $\{\boldsymbol{e}(\tau_{2i-1,d}), \boldsymbol{e}(\tau_{2i,d})\}_{i\in[k]}$ are linearly independent for any $d' \in [D]$, then the balanced condition is sufficient to show existence.*

Remark: Recall that $\mathcal{P}_\perp$ projects to the subspace orthogonal $\mathbf{1}\mathbf{1}^\top$. The assumption in the necessary condition can be intuitively understood as requiring all tokens to have nonzero contributions to the prediction, because otherwise $W_V^{(2)}\boldsymbol{e}(\tau_{t,d})$ will not contribute to prediction after the LayerNorm.

Proof. **Necessity of the balanced condition.** By Equation (1), the attention output is directly used as the input of LayerNorm, thus we *ignore the normalization* from the softmax operation. For any prefix p ending with a closed bracket $\tau_{2j,d}$ for $d \geq 1$

and containing brackets of all depths in $[D]$, let p_m be the prefix obtained by inserting m pairs of $\{\tau_{2i-1,d'}, \tau_{2i,d'-1}\}$ for arbitrary $i \in [k]$ and depth $d' \in [D]$. Denote the projection of the unnormalized attention output by

$$u(\tau_{t_1,d_1}, \tau_{t_2,d_2}) := \mathcal{P}_\perp \exp\left(\boldsymbol{e}\big(\tau_{t_1,d_1}\big)^\top (W_K^{(2)})^\top W_Q^{(2)} \boldsymbol{e}\big(\tau_{t_2,d_2}\big)\right) W_V^{(2)} \boldsymbol{e}\big(\tau_{t_1,d_1}\big). \tag{13}$$

Then, by Equation (2), we have,

$$\mathcal{T}(p_m) = g^{(2)}\left(\mathrm{LN}^{(2)}\left(v + m\left(u(\tau_{2j,d}, \tau_{2i,d'-1}) + u(\tau_{2j,d}, \tau_{2i-1,d'})\right)\right) + \boldsymbol{e}(\tau_{2j,d})\right), \tag{14}$$

where v denotes the unnormalized second-layer output given p as input.

Towards reaching a contradiction, suppose $u(\tau_{2j,d}, \tau_{2i,d'}) + u(\tau_{2j,d}, \tau_{2i-1,d'+1}) \neq 0$. Based on the continuity of the projection function and the LayerNorm Layer, we can show that $\lim_{m\to\infty} \mathcal{T}(p_m)$ depend only on grammar depths d, d' and types $2j, 2i-1, 2i$, which, however, are not sufficient to determine the next-token probability from p_m, since the latter depends on the type of the last unmatched open bracket in p. This contradicts the assumption that the model achieves the Bayes-optimal loss for any length N. Hence we must have

$$u(\tau_{2j,d}, \tau_{2i,d'-1}) + u(\tau_{2j,d}, \tau_{2i-1,d'}) = 0. \tag{15}$$

Finally, since we assume $\mathcal{P}_\perp W_V^{(2)} \boldsymbol{e}(\tau_{t,d}) \neq 0$, we conclude that

$$\left(\boldsymbol{e}(\tau_{2i-1,d'}) - \boldsymbol{e}(\tau_{2i,d'-1})\right)^\top (W_K^{(2)})^\top W_Q^{(2)} \boldsymbol{e}(\tau_{2j,d}) = \ln\left(\frac{\|\mathcal{P}_\perp W_V \boldsymbol{e}(\tau_{2i-1,d'})\|_2}{\|\mathcal{P}_\perp W_V \boldsymbol{e}(\tau_{2i,d'-1})\|_2}\right).$$

Note that the right hand side is independent of j, d. This concludes the proof for the necessity of the condition.

Sufficiency of the balance condition. We will show a construction, using the embedding function $\boldsymbol{e}(\tau_{t,d})$ as given in Equation (Type 1). Fix any $j \in [k], d \in [D]$. By Equation (12), we can assume that there exists an $a \in \mathbb{R}^{k\times D}$ such that for $i \in [k]$, $d', d \in [D]$, it satisfies

$$a_{i,d'} \triangleq \left(\boldsymbol{e}(\tau_{2i-1,d'}) - \boldsymbol{e}(\tau_{2i,d'-1})\right)^\top (W_K^{(2)})^\top W_Q^{(2)} \boldsymbol{e}(\tau_{2j,d}).$$

We can then choose $W_V^{(2)}$ for $i \in [k]$ and $d' \in [D]$ such that

$$\begin{aligned} W_V^{(2)} \boldsymbol{e}(\tau_{2i,d'-1}) &= -\exp(a_{i,d'}) \boldsymbol{o}_{(2i-1)\times(D-1)+d'}. \\ W_V^{(2)} \boldsymbol{e}(\tau_{2i-1,d'}) &= \boldsymbol{o}_{(2i-1)\times(D-1)+d'}. \end{aligned} \tag{16}$$

Such $W_V^{(2)}$ is guaranteed to exist: solving for $W_V^{(2)}$ is equivalently to solving the linear equation $W_V^{(2)} \boldsymbol{E} = \boldsymbol{O}$, where $\boldsymbol{E}, \boldsymbol{O} \in \mathbb{R}^{2kD\times 2kD}$ are defined according to Equation (16) [4] and $\boldsymbol{E}$ is of full rank by the linear independence assumption.

It can be checked that choosing $W_V^{(2)}$ to satisfy Equation (16) will also make Equation (15) satisfied. Hence for any prefix p of length n ending with a closed bracket $\tau_{2j,d}$ satisfying $d \geq 1$, suppose the list of unmatched open brackets in p is $[\tau_{2j_1-1,1}, \tau_{2j_2-1,2}, \ldots, \tau_{2j_m-1,d}]$, then suppose X is the input of the second layer, we will have the last column (i.e. corresponding to the last position) of the input to the LayerNorm satisfies,

$$W_V^{(2)} X \cdot \left[\sigma\left(\mathcal{C} \cdot \frac{(W_K^{(2)} X)^\top (W_Q^{(2)} X)}{\sqrt{d_a}}\right)\right]_{:,n} = \sum_{s=1}^{d} u(\tau_{2j_s-1,s}, \tau_{2j,d}), \tag{17}$$

where $\mathcal{C}$ denotes the causal mask.

Finally we can choose the weights in the LayerNorm to be sufficiently small such that the largest index of the last column of input to $g^{(2)}$ is determined by $X_{:,n}$. This weights can always be chosen because the norm of the output of LayerNorm is bounded by 1 and $\boldsymbol{e}(\tau_{t,d})$ are linearly independent, hence nonzero. Then the next token probability can be determined by:

[4] Specifically, $\boldsymbol{E} = [\boldsymbol{e}(\tau_{1,1}), \boldsymbol{e}(\tau_{1,2}), \cdots, \boldsymbol{e}(\tau_{2k,D-2}), \boldsymbol{e}(\tau_{2k,D-1})]$, i.e. $\boldsymbol{E}$ is the collection of all $\boldsymbol{e}(\tau_{t,d})$. $\boldsymbol{O}$ is defined such that for every d', $\boldsymbol{O}_{:,t(D-1)+d'} = -\exp(a_{t/2,d'}) \boldsymbol{o}_{(t-1)(D-1)+d'}$ if t is even, and $\boldsymbol{O}_{:,t(D-1)+d'} = \boldsymbol{o}_{t(D-1)+d'}$ if t is odd.

1. The last bracket in p, when p ends with an open bracket or a closed bracket with depth 0,

2. The type of last unmatched open bracket in p: suppose the grammar depth of this unmatched open bracket is d, then we only need to look at indices $(2i-1)\times(D-1)+d$ for $i \in [k]$. Among values of these indices, if the value is maximized at $i' \in [k]$, then the correct type of the unmatched bracket is i'.

To complete the proof, note that the above functionality can be implemented with a combination of feedforward layers. Specifically, since there are only a finite number of possible input to g, we can construct a 2-layer ReLU network that memorize the values for all inputs, which requires a width that is polynomial in the number of possible inputs. □

D.2.1. Proof of Corollary 3.3

Corollary D.3 (Corollary 3.3, restated)**.** *There exists a two-layer Transformer with uniform attention and without position embedding (but with causal mask) that can generate the Dyck language of arbitrary length.*

Proof. It is easy to see that the condition in Theorem 3.2 is satisfied. Hence it suffices to construct a uniform attention first layer that can generate the embedding in Equation (Type 1). Let $W_V^{(1)}$ be the identity matrix, and suppose Z is the one-hot embeddings of a prefix p of length n, where each token of type t for $t \in [2k]$ is encoded as $\boldsymbol{o}_t$. Then, the last column of Z satisfies

$$W_V^{(1)} Z\Big[\sigma\Big(\mathcal{C}\cdot\frac{(W_K^{(1)}Z)^\top(W_Q^{(1)}Z)}{\sqrt{d_a}}\Big)\Big]_{:,n} = \sum_{i=1}^{2k}\#\{\text{token of type } t \text{ in } p\}\boldsymbol{o}_t. \tag{18}$$

where $\mathcal{C}$ denotes the causal mask.

The depth of the n_{th} token can then be determined by counting the number of i satisfying the value of index $2i-1$ and $2i$ in the last column of Z are different by 1. Similar to the proof of Theorem D.2, this function can be implemented with a combination of feedforward layers and LayerNorm layers and the proof is then completed. □

D.3. Proof of Theorem C.1

Let's first define a quantity for convenience of later exposition. Let u be defined as in Equation (13). For any $i \in [k], d \in [D]$ and $\tilde{\boldsymbol{t}} \in [k]^{d-1}$, denote the quantity

$$Q(i,d,\tilde{\boldsymbol{t}}) := \sum_{1\le d'<d} u(\tau_{2i,d-1}, \tau_{2\tilde{\boldsymbol{t}}_{d'}-1,d'}) + u(\tau_{2i,d-1}, \tau_{2i-1,d}) + u(\tau_{2i,d-1}, \tau_{2i,d-1}), \tag{19}$$

where $\tilde{\boldsymbol{t}}_{d'}$ denotes the d'_{th} entry of $\tilde{\boldsymbol{t}}$. That is, $\tilde{\boldsymbol{t}}$ is a string of $d-1$ open brackets. Let τ_i denote a bracket of type $i \in [2k]$ without specifying the grammar depth (i.e. the grammar depth is implicit from the context), then $Q(i,d,\tilde{\boldsymbol{t}})$ can be considered as the unnormalized output of the second-layer attention of a Transformer on the input sequence $\tilde{\boldsymbol{t}} \oplus \tau_{2i-1}\tau_{2i}$ [5].

Theorem D.4 (Approximate Balance; formal version of Theorem C.1)**.** *Consider a two-layer Transformer $\mathcal{T}$ with a minimal first layer trained with the mean squared error (Equation (11)). For any $\gamma, N > 0$ and sufficiently small ϵ, suppose $\mathrm{g}^{(2)}$ is γ-Lipschitz, and suppose the set of second-layer weights $\bar{\theta}_N^{(2)}$ satisfies that $\mathcal{L}(\mathcal{T}[\bar{\theta}_N^{(2)}], \mathcal{D}_{q,k,D,N}) \le q^{-N}\epsilon$. Then, there exists a constant $C_{\gamma,\epsilon,D}$, such that for any $0 \le d' \le D, 1 \le d \le D, i,j \in [k]$, it holds that*

$$S_{d,d',i,j}[\bar{\theta}_N^{(2)}] \le \frac{C_{\gamma,\epsilon,D}}{N} P_{d,j}[\bar{\theta}_N^{(2)}]. \tag{20}$$

where

$$S_{d,d',i,j}[\bar{\theta}^{(2)}] = \Big\| u(\tau_{2j,d}, \tau_{2i,d'}) + u(\tau_{2j,d}, \tau_{2i-1,d'+1}) \Big\|_2, \tag{21}$$

$$P_{d,j}[\bar{\theta}^{(2)}] = \min_{\boldsymbol{t}'\in[k]^{d-1}, \boldsymbol{t}'_d \ne \boldsymbol{t}_d} \|Q(i,d,\boldsymbol{t}')\|_2, \tag{22}$$

[5] $s \oplus t$ denotes the concatenation of two strings s, t, same as in Equation (Type 1)-(Type 3). The concatenation of two tokens τ_i, τ_j is simply written as $\tau_i\tau_j$.

for $\boldsymbol{t} = \arg\min_{\boldsymbol{t}' \in [k]^{d-1}} \|Q(2j, d, \boldsymbol{t}')\|_2$. [6]

Proof. The key idea is similar to the proof of necessity in Theorem 3.2. That is, we will construct two input sequences with different next-word distributions, and show that the approximate balance condition must hold so that inserting (a bounded number of) pairs of matching brackets does not collapse the two predicted distributions given by the Transformer.

Constructing the input sequences.

Let $\boldsymbol{t} := \arg\min_{\tilde{\boldsymbol{t}} \in [k]^{d-1}} \|Q(2j, d, \tilde{\boldsymbol{t}})\|_2$, and let $\boldsymbol{t}'$ denote the prefix that minimizes $\|Q(2j, d, \tilde{\boldsymbol{t}})\|_2$ subject to the constraint that $\boldsymbol{t}'$ must differ from $\boldsymbol{t}$ in the last (i.e. $(d-1)_{th}$) position, i.e.

$$\boldsymbol{t}' = \arg \min_{\tilde{\boldsymbol{t}}' \in [k]^{d-1}, \boldsymbol{t}'_{d-1} \neq \boldsymbol{t}_{d-1}} Q(2j, d, \tilde{\boldsymbol{t}}').$$

The motivation for such choices of $\boldsymbol{t}, \boldsymbol{t}'$ is that since they differ at least by the last position which is an open bracket, they must lead to different next-word distributions. Note also that $P_{d,j}[\bar{\theta}^{(2)}] = \|Q(2j, d, \boldsymbol{t}')\|$.

With the above definition of $\boldsymbol{t}, \boldsymbol{t}'$, consider two valid Dyck prefixes p_1 and p_2 with length no longer than N, defined as follows: for any $d, d' \in [D], i, j \in [k]$, consider a common prefix $p = \underbrace{\tau_{2i-1} \dots \tau_{2i-1}}_{d' \text{ open brackets}} \underbrace{\tau_{2i-1}\tau_{2i} \dots \tau_{2i-1}\tau_{2i}}_{\lfloor \frac{N-2d'-2d}{2} \rfloor \text{ pairs}} \underbrace{\tau_{2i} \dots \tau_{2i}}_{d' \text{ closed brackets}}$, and set:

$$p_1 = p \oplus \boldsymbol{t} \oplus \tau_{2j-1}\tau_{2j},$$
$$p_2 = p \oplus \boldsymbol{t}' \oplus \tau_{2j-1}\tau_{2j}.$$

In the following, we will show that the approximate balance condition must hold for the predictions on p_1, p_2 to be sufficiently different.

Bounding the difference in Transformer outputs. The Transformer outputs on p_1, p_2 satisfies

$$\|\mathcal{T}[\bar{\theta}_N^{(2)}](p_1) - \mathcal{T}[\bar{\theta}_N^{(2)}](p_2)\|_2 \geq 1 - \text{TV}(p_1, p_2) - o_\epsilon(1) = \Omega(1), \tag{23}$$

where $\text{TV}(p_1, p_2)$ denotes the TV distance in the next-word distributions from p_1 and p_2, and $o_\epsilon(1)$ means the term will go to zero for sufficiently small ϵ. The former is bounded by the construction of p_1, p_2. The latter is bounded because of the assumption on $\bar{\theta}_N^{(2)}$, which states that the set of second-layer weights $\bar{\theta}_N^{(2)}$ satisfies that $\mathcal{L}(\mathcal{T}[\bar{\theta}_N^{(2)}], \mathcal{D}_{q,k,D,N}) \leq q^{-N}\epsilon$ with sufficiently small ϵ.

Define by A_p the contribution of p to the attention output (before LayerNorm) of the last position of p_1, p_2, i.e.

$$\begin{aligned} A_p = &\sum_{0 \leq d'' < d'} \left(u(\tau_{2j,d-1}, \tau_{2i,d''}) + u(\tau_{2j,d-1}, \tau_{2i-1,d''+1})\right) \\ &+ \lfloor \frac{N - 2d' - 2d}{2} \rfloor \left(u(\tau_{2j,d-1}, \tau_{2i,d'}) + u(\tau_{2j,d-1}, \tau_{2i-1,d'+1})\right). \end{aligned} \tag{24}$$

The attention outputs (before LayerNorm) of p_1, p_2, denoted by $A(p_1)$ and $A(p_2)$, satisfy that

$$\begin{aligned} \mathcal{P}_\perp A(p_1) &= \mathcal{P}_\perp(A_p + Q(2j, d, \boldsymbol{t})), \\ \mathcal{P}_\perp A(p_2) &= \mathcal{P}_\perp(A_p + Q(2j, d, \boldsymbol{t}')). \end{aligned} \tag{25}$$

Note that for any prefix p', $\mathcal{T}[\bar{\theta}_N^{(2)}](p') = \mathrm{g}^{(2)}(\mathcal{P}_\perp A(p'))$. Then, since $\mathrm{g}^{(2)}$ is γ-Lipschitz,

$$\left\| \frac{\mathcal{P}_\perp A(p_1)}{\|\mathcal{P}_\perp A(p_1)\|_2} - \frac{\mathcal{P}_\perp A(p_2)}{\|\mathcal{P}_\perp A(p_2)\|_2} \right\|_2 \geq \frac{1 - \text{TV}(p_1, p_2) - O_\epsilon(1)}{\gamma} = \Omega_{\gamma,\epsilon}(1). \tag{26}$$

We show that A_p should not be too much larger in norm than $Q(2j, d, \boldsymbol{t})$ or $Q(2j, d, \boldsymbol{t}')$. First let's state a helper lemma about the contrapositive:

[6] *Erratum*: This definition of $P_{d,j}[\theta^{(2)}]$ is slightly different from the one in the original main paper submitted on May 17th. The definition here and in the current main paper have been corrected.

Lemma D.5. *For any $\epsilon > 0$, there exists a constant R_ϵ, such that for any $a, b \in \mathbb{R}^d$ and any $r \in \mathbb{R}^d$ such that $\|r\|_2 \geq R_\epsilon \cdot \max\{\|a\|_2, \|b\|_2\}$, it holds that*

$$\left\| \frac{a+r}{\|a+r\|_2} - \frac{b+r}{\|b+r\|_2} \right\|_2 \leq \epsilon.$$

Proof. Denote $r_0 := \max\{\|a\|_2, \|b\|_2\}$. Then $R_\epsilon := \frac{4r_0}{\epsilon} + 1$ suffices:

$$\begin{aligned} &\left\| \frac{r+a}{\|r+a\|_2} - \frac{r+b}{\|r+b\|_2} \right\| \leq \|r\| \cdot \left| \frac{1}{\|r+a\|} - \frac{1}{\|r+b\|} \right| + \frac{\|a\|}{\|r+a\|} + \frac{\|b\|}{\|r+b\|} \\ \leq &\|r\| \cdot \left(\frac{1}{\|r\| - r_0} - \frac{1}{\|r\| + r_0} \right) + \frac{2r_0}{\|r\| - r_0} \\ = &\frac{2r_0}{\|r\| - r_0} \cdot \left(\frac{\|r\|}{\|r\| + r_0} + 1 \right) \leq \frac{4r_0}{\|r\| - r_0} \leq \frac{4r_0}{R_\epsilon - r_0} \leq \epsilon. \end{aligned}$$

□

Lemma D.5 implies that if A_p is too large, then the output on p_1, p_2 (Equation (26)) won't be sufficiently different. Let $P_{d,j}[\bar{\theta}_N^{(2)}]$ be defined as in Equation (21) and let R_ϵ be the constant in Lemma D.5, we need to bound $\|\mathcal{P}_\perp A_p\|$ by

$$\|\mathcal{P}_\perp A_p\|_2 \leq R_\epsilon \|P_{d,j}[\bar{\theta}_N^{(2)}]\|_2. \tag{27}$$

As Equation (27) holds for p with any d, d', by an induction on d' (from 1 to d) on the second term in Equation (24), one can show that there exists C (depending on R_ϵ), such that,

$$S_{d,d',i,j} = \|u(\tau_{2j,d-1}, \tau_{2i,d-1}) + u(\tau_{2j,d-1}, \tau_{2i-1,d-1})\| \leq \frac{C}{N} \|P_{d,j}[\bar{\theta}_N^{(2)}]\|_2. \tag{28}$$

The proof of Equation (28) can be carried out inductively over d from 1 to D. □

Proof of Corollary C.2. This proof is in fact a direct combination of Theorems 3.2 and C.1. By Theorem 3.2 we know there exists a weight $\theta^{(2)*}$ that can reach zero loss for arbitrarily length N. Then it holds that $\|\theta_{\lambda,N}\|_2 \leq \|\theta^*\|$ as $\theta_{\lambda,N}$ minimizes the regularized loss. Notice bounded weight implies bounded lipschitzness of $\mathrm{g}^{(2)}$, The rest follows as Theorem C.1. □

D.4. Proof of Theorem 3.4 – Indistinguishability from a single component

We now show the limitation of interpretability from a single component, using a Lottery-Ticket-style argument by pruning from large random Transformers.

For this section only, we will make the following modifications to the Transformer architecture in (2):

- We lower bound the normalization factor in the LayerNorm by some constant C, namely we consider:

  $$\mathrm{LN}_C(x) = \frac{\mathcal{P}_\perp x}{\max\{\|\mathcal{P}_\perp x\|_2, C\}},$$

  We need this assumption for technical reasons (to make the LayerNorm Lipschitz). We note that thresholding at C is also a common practice empirically due to numerical stability concerns.

- We assume all affine layers and linear head in the Transformer have zero bias. This is mainly for technical convenience, and was also assumed in prior works on theoretical analysis of the lottery ticket hypothesis (Pensia et al., 2020). Note that this is not a restriction since bias can be removed with homogeneous coordinates.

We will also consider a modified projection function $\mathrm{g}_{\mathrm{large}}^{(l)}$ consisting of a 4-layer MLP, which will be used in the to-be-pruned large random Transformers:

$$\mathrm{g}_{\mathrm{large}}(x) = \mathrm{LN}\left(W_4 \mathrm{ReLU}\left(W_3 \mathrm{ReLU}\left(W_2 \mathrm{ReLU}\left(W_1 x\right)\right)\right)\right) + x, \tag{29}$$

where $W_1, W_4^\top \in \mathbb{R}^{w_{\text{large}} \times m_{\text{large}}}$, $W_2, W_3 \in \mathbb{R}^{w_{\text{large}} \times w_{\text{large}}}$, for some $w_{\text{large}}, m_{\text{large}}$.

We are now ready to state the main theorem of this section:

Theorem D.6 (Indistinguishability From a Single Component (Theorem 3.4 restated)). *Consider a L-layer Transformer $\mathcal{T}$ with embedding dimension m, width w and $g^{(k)}(x) = \mathrm{LN}_C\left(W_2^{(k)} \mathrm{ReLU}\left(W_1^{(k)} x\right)\right) + x$. Suppose $\|W\|_2 = O(1)$ for every weight matrix W in $\mathcal{T}$. For $\delta \in (0,1)$, consider a larger* random *Transformer $\mathcal{T}_{large}$ with $4L$ layers, embedding dimension $m_{\text{large}} = O(d \log(d/\delta))$, and width $w_{\text{large}} = O(\max\{m, w\}\} \log \frac{wmLN}{\epsilon\delta})$, and projection function g_{large}, whose weights are randomly sampled as $W_{i,j} \sim U(-1,1)$ for every $W \in \mathcal{T}_{large}$.*

Then, with probability $1-\delta$ over the randomness of $\mathcal{T}_{large}$, we can obtain a nonstructural pruning (Definition 2.1) of $\mathcal{T}_{large}$, denoted as $\mathcal{T}'_{large}$, which ϵ-approximates $\mathcal{T}$. That is, $\forall \boldsymbol{X} \in \mathbb{R}^{m \times N}$ with $\|\boldsymbol{X}_{:,i}\|_2 \leq 1,\ \forall i \in [N]$,

$$\|\mathcal{T}'_{large}(\boldsymbol{X}) - \mathcal{T}(\boldsymbol{X})\|_2 \leq \epsilon.$$

Moreover, pick any weight matrix W in $\mathcal{T}_{large}$, with probability $1-\delta$, for any smaller Transformers $\mathcal{T}_1, \mathcal{T}_2$ satisfying same conditions as $\mathcal{T}$, we have two pruned Transformers $\mathcal{T}_{Large,1}, \mathcal{T}_{Large,2}$ based on $\mathcal{T}_{large}$, such that they coincide on the pruned weight of W, and $\mathcal{T}_{Large,i}$ ϵ-approximate $\mathcal{T}_i$, $\forall i \in \{1,2\}$.

Proof. We will first introduce some notation. For vector $x \in \mathbb{R}^a$ and $y \in \mathbb{R}^b$, we will use $x \oplus y$ to denote their concatenation. We will use 0^a to denote the all-zero vector with dimension a. We will also assume without loss of generality that $w \geq 2d$. [7]

In the following, a *random network* refers to a network whose weights have entries sampled from a uniform distribution, i.e. $W_{i,j} \sim U(-1,1)$ for every weight W in the random network.

We will first recall Lemma D.7 from (Pensia et al., 2020) which shows that a pruned 2-layer random network can approximate a linear function.

Lemma D.7 (Theorem 1 of (Pensia et al., 2020)). *Let $W \in \mathbb{R}^{d' \times d}$, $\|W\|_2 = O(1)$, then for $\sigma \in \{\mathrm{ReLU}, \mathcal{I}\}$, for a random network $g(x) = W_2 \sigma(W_1 x)$ with $W_2 \in \mathbb{R}^{d' \times h}, W_1 \in \mathbb{R}^{h \times d}$ for hidden dimension $h = O(d \log(\frac{dd'}{\min\{\epsilon, \delta\}}))$, with probability $1-\delta$, there exists boolean matrices M_1, M_2, such that for any $x \in \mathbb{R}^d$, $\|x\|_2 = O(1)$,*

$$\|(M_2 \odot W_2)\sigma\big((M_1 \odot W_1)x\big) - Wx\| \leq \epsilon,$$

where $\odot$ denotes the Hadamard product.

We will use the following helper lemma:

1. A pruned 4-layer projection function of a Transformer layer can approximate a 2-layer ReLU network applied to each token (Lemma D.8).

2. A pruned random Transformer layer can approximate a linear function applied independently to each token (Lemma D.9).

3. Two pruned random Transformer layers can approximate a fixed smaller Transformer layer. (Lemma D.12)

We can now prove the theorem.

To show ϵ-approximation, we can prune the large Transformer to approximate the smaller Transformer layer by layer by Lemma D.12. The linear head $W^{(head)}$ can be pruned using Lemmas D.9 and D.11, and combined with one layer of the Transformer, the linear head of the smaller Transformer can be approximated.

Further, as we only need 2 layers to approximate one layer of the smaller Transformer, for an arbitrary layer l, we can prune the layer l of the large Transformer to ϵ-approximate identity function. This then concludes the proof for indistinguishability from single components. □

[7] We can always pad dimensions if w is too small.

D.4.1. Helper lemmas for Theorem D.6

We first show that a pruned 4-layer projection function in a Transformer layer can approximate a 2-layer ReLU network applied to each token:

Lemma D.8. *Under the condition of Theorem D.6, for any two matrices $W_1 \in \mathbb{R}^{d\times w}, W_2 \in \mathbb{R}^{w\times d}$, $\|W_1\|_2, \|W_2\|_2 = O(1)$, for any $\delta \in (0,1)$ and $l \in [4L]$, with probability $1-\delta$, there exists an unstructured pruning of $\mathrm{g}_{\text{large}}^{(l)}$, $\mathrm{g}_{\text{large}}^{(l)'}$, satisfying that $\forall \boldsymbol{X} \in \mathbb{R}^{m\times N}$ with $\|\boldsymbol{X}_{:,i}\|_2 = O(1),\ \forall i \in [N]$,*

$$\forall \boldsymbol{R} \in \mathbb{R}^{(m_{\text{large}}-m)\times N}, \left\| \left(\mathrm{g}_{\text{large}}^{(l)'} \left(\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{R} \end{bmatrix} \right) \right)_{1:m,:} - W_2 \mathrm{ReLU}\left(W_1 \boldsymbol{X} \right) \right\|_2 \le \epsilon,$$

where $M_{1:m,:}$ denotes the first m rows of a matrix M.

Proof. Recall the definition of the projection function of a Transformer layer is

$$\mathrm{g}_{\text{large}}^{(l)}(x) = \mathrm{LN}\left(W_4^{(l)} \mathrm{ReLU}\left(W_3^{(l)} \mathrm{ReLU}\left(W_2^{(l)} \mathrm{ReLU}\left(W_1^{(l)} x \right)\right)\right)\right) + x.$$

We will prune the LayerNorm by setting it to the identity. Now we only need to show that there exists boolean matrices M_1, M_2, M_3, M_4, such that,

$$\Big\| \Big(M_4 \odot W_4^{(l)} \mathrm{ReLU}\big((M_3 \odot W_3^{(l)}) \mathrm{ReLU}\big((M_2 \odot W_2^{(l)}) \mathrm{ReLU}\big((M_1 \odot W_1^{(l)}) \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{R} \end{bmatrix} \big)\big)\big)\Big)_{1:m,:} \\ - W_2 \mathrm{ReLU}\left(W_1 \boldsymbol{X} \right) - \boldsymbol{X} \Big\|_2 \le \epsilon.$$

We can first choose

$$(M_1)_{:,(m+1,\dots,m_{\text{large}})} = 0, (M_4)_{(m+1,\dots,m_{\text{large}}),:} = 0,$$
$$(M_2)_{(w+2m+1,\dots,w_{\text{large}}),:} = 0, (M_3)_{:,(w+2m+1\dots,w_{\text{large}})} = 0$$

Then by Lemma D.7, there exists boolean matrices M_1, M_2, M_3, M_4 satisfying previous constraint, such that,

$$\left\| \left((M_2 \odot W_2^{(l)}) \mathrm{ReLU}\left((M_1 \odot W_1^{(l)}) \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{R} \end{bmatrix} \right) \right)_{1:w+2m} - \begin{bmatrix} W_1 \\ \mathcal{I} \\ -\mathcal{I} \end{bmatrix} \boldsymbol{X} \right\| \le \frac{\epsilon}{4}.$$

$$\forall \boldsymbol{X'} \in \mathbb{R}^{(w+2m)\times N}, \left\| (M_4 \odot W_4^{(l)}) \mathrm{ReLU}\big((M_3 \odot W_3^{(l)}) \begin{bmatrix} \boldsymbol{X'} \\ \boldsymbol{R'} \end{bmatrix} \big) - \begin{bmatrix} W_2 & \mathcal{I} & -\mathcal{I} \end{bmatrix} \boldsymbol{X'} \right\| \le \frac{\epsilon}{4} \cdot \frac{\max_{i\in[N]} \|\boldsymbol{X'}_{:,i}\|_2}{\|W_1\|_2}.$$

This then concludes the proof. $\square$

Based on the above lemma, we can prove that a pruned Transformer layer can approximate a linear function applied independently to each token.

Lemma D.9. *Under the conditions in Theorem D.6, for any matrix $W \in \mathbb{R}^{m\times m}, \|W\|_2 = O(1)$, $\delta \in (0,1)$ and $l \in [4L]$, with probability $1-\delta$, there exists an unstructured pruning of $\mathcal{T}_{\mathit{large}}^{(l)}$, $\mathcal{T}_{\mathit{large}}^{(l)'}$, satisfying that $\forall \boldsymbol{X} \in \mathbb{R}^{m\times N}$ with $\|\boldsymbol{X}_{:,i}\|_2 = O(1),\ \forall i \in [N]$, we have*

$$\forall \boldsymbol{R} \in \mathbb{R}^{(m_{\text{large}}-m)\times N}, \left\| \left(\mathcal{T}_{\mathit{large}}^{(l)'} (\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{R} \end{bmatrix}) \right)_{1:m,:} - W \boldsymbol{X} \right\|_2 \le \epsilon.$$

Proof. Recall that given an input $\boldsymbol{X'}$, a Transformer layer computes $\mathcal{T}_{\text{large}}^{(l)}(\boldsymbol{X'}) = \mathrm{g}_{\text{large}}^{(l)}\Big(\mathrm{LN}\Big(W_V^{(l)} \boldsymbol{X'} \mathrm{Attn}(\boldsymbol{X'}) \Big) + \boldsymbol{X'} \Big)$, where $\mathrm{Attn}(\boldsymbol{X'}) := \sigma\Big(\mathcal{C} \cdot \frac{(W_K^{(l)} \boldsymbol{X'})^\top (W_Q^{(l)} \boldsymbol{X'})}{\sqrt{d_a}} \Big)$ computes the attention pattern. Lemma D.8 already shows that $\mathrm{g}_{\text{large}}^{(l)}$ can approximate a linear transformation; it remains to show that the linear transformation can compute $W\boldsymbol{X}$.

We can first choose two matrices $W_1 \in \mathbb{R}^{w\times m}, W_2 \in \mathbb{R}^{m\times w}$ satisfying that

$$W_1 = [\mathcal{I}_m, -\mathcal{I}_m, 0^{m\times(w-2m)}]^\top.$$
$$W_2 = [W, -W, 0^{m\times(w-2m)}]$$

Then we have that $\|W_1\|_2, \|W_2\|_2 = O(1)$ and $W_2\mathrm{ReLU}(W_1\boldsymbol{X}) = W\boldsymbol{X}$. We can then turnoff the LayerNorm after the attention module and prune W_V to be 0, which effectively removes the effect of attention and rely solely on the residual link. The proof can now be completed by applying Lemma D.8. ☐

We will then show that two pruned Transformer layers can approximate a fixed smaller Transformer layer. The key technical difficulty is approximating the attention module and bounding the error of the approximation after LayerNorm. We will first show a lemma showing the Lipschitzness of the LayerNorm (with cutoff at some constant C).

Lemma D.10. *For LayerNorm function defined as $\mathrm{LN}(x) = \frac{\mathcal{P}_\perp x}{\max\{\|\mathcal{P}_\perp x\|_2, C\}}, x \in \mathbb{R}^m$, there exists constant C_1 depending on C, such that for any $x, y \in \mathbb{R}^m$, it holds that,*

$$\left\|\mathrm{LN}(x) - \mathrm{LN}(y)\right\|_2 \le C_1\|x-y\|_2.$$

Proof. We will proceed by a case analysis:

1. If $\|\mathcal{P}_\perp x\|_2, \|\mathcal{P}_\perp y\|_2 \le C$, then $\left\|\mathrm{LN}(x) - \mathrm{LN}(y)\right\|_2 = \frac{\|\mathcal{P}_\perp x - \mathcal{P}_\perp y\|_2}{C} \le \frac{1}{C}\|x-y\|_2$.

2. If $\|\mathcal{P}_\perp x\|_2, \|\mathcal{P}_\perp y\|_2 > C$, then $\left\|\mathrm{LN}(x) - \mathrm{LN}(y)\right\|_2 = \frac{\|\mathcal{P}_\perp x - \mathcal{P}_\perp y\|_2}{\|\mathcal{P}_\perp y\|_2} + \left|1 - \frac{\|\mathcal{P}_\perp x\|_2}{\|\mathcal{P}_\perp y\|_2}\right| \le \frac{2}{C}\|x-y\|_2$.

3. If $\|\mathcal{P}_\perp x\|_2 < C$ and $\|\mathcal{P}_\perp y\|_2 > C$, then $\left\|\mathrm{LN}(x) - \mathrm{LN}(y)\right\|_2 = \frac{\|\mathcal{P}_\perp x - \mathcal{P}_\perp y\|_2}{\|\mathcal{P}_\perp y\|_2} + \left|\frac{\|\mathcal{P}_\perp x\|_2}{C} - \frac{\|\mathcal{P}_\perp x\|_2}{\|\mathcal{P}_\perp y\|_2}\right| \le \frac{2}{C}\|x-y\|_2$.

The cases exhaust all possibilities, thus the proof is completed. ☐

We also need to show there exists a pruning of the value matrix in $\mathcal{T}_{\text{large}}$ such that it has eigenvalues with magnitude $\Theta(1)$.

Lemma D.11. *For a matrix $W \in \mathbb{R}^{w_{\text{large}}\times w_{\text{large}}}$, with probability at least $1-\delta$, there exists a pruning of W, named W', such that all the nonzero entries is contained in a $d\times d$ submatrix of W' that satisfies that (1) all its eigenvalues are within $(\frac{1}{2}, 1)$, (2) the index of row specifying the submatrix and the index of column specifying the submatrix are disjoint.*

Proof. As $w_{\text{large}} = \Omega(m\log(\frac{d}{\delta}))$, hence we can split $W_{1:\lceil m_{\text{large}}/2\rceil, \lceil m_{\text{large}}/2\rceil+1:m_{\text{large}}}$ into $(m \times (m$ blocks, each with width at least $O(\log(\frac{(m}{\delta}))$ [8]. Within each block, with probability $1-\frac{\delta}{(m}$, there exists at least one entry that has value at least $\frac{1}{2}$. We can then choose d disjoint entries in W that are all at least $\frac{1}{2}$, indexed with $\{(a_i, b_i)\}_{i\in[d]}$ where $a_i < a_j$ and $b_i < b_j$ for $i < j$. We can then prune all other entries to zero. Consider the submatrix defined by entries (a, b) for $a \in \{a_i\}_{i\in m}$ and $b \in \{b_i\}_{i\in m}$. Then, this submatrix will be diagonal and contains eigenvalues within $(\frac{1}{2}, 1)$. Further $\{a_i\}_{i\in m}$ and $\{b_i\}_{i\in m}$ must be disjoint because $a_i \le \lceil m_{\text{large}}/2\rceil < b_i$. The proof is then completed. ☐

Next, we show that two random Transformer layers can be pruned to approximate a given Transformer layer.

Lemma D.12. *Under the condition of Theorem 3.4, for any matrix $W \in \mathbb{R}^{d\times d}$, $\|W\|_2 = O(1)$, $\delta \in (0,1)$ and $t \in [4L]$, for any $l \in [L]$, with probability $1-\delta$, there exists an unstructured pruning of $\mathcal{T}^{(t)}_{large}, \mathcal{T}^{(t+1)}_{large}$, named $\mathcal{T}^{(t)'}_{large}, \mathcal{T}^{(t+1)'}_{large}$, satisfying that $\forall \boldsymbol{X} \in \mathbb{R}^{d\times N}$ with $\|\boldsymbol{X}_{:,i}\|_2 = O(1)$, $\forall i \in [N]$,*

$$\forall \boldsymbol{R} \in \mathbb{R}^{(m_{\text{large}}-m)\times N}, \|\mathcal{T}^{(t+1)'}_{large}\left(\mathcal{T}^{(t)'}_{large}\left([\boldsymbol{X}_{:,i} \oplus \boldsymbol{R}_{:,i}]_{i\in[N]}\right)\right)_{1,\ldots,m} - \mathcal{T}^{(l)}(\boldsymbol{X})\|_2 \le \epsilon.$$

Proof. We will prune the larger transformer in the following order.

[8] $O(\cdot)$ hides absolute constants arising from the change of basis in the logarithm.

1. We will prune $W_V^{(t+1)}$ according to Lemma D.11 and name the pruned matrix $W_V^{(t+1)'}$. By Lemma D.11, all the nonzero entries is contained in a $d \times d$ submatrix of W' that satisfies that all its eigenvalues are within $(\frac{1}{2}, 1)$. We will prune $W_V^{(t+1)}$ in this way, named $W_V^{(t+1)'}$ and assume WLOG the submatrix is the one specified by row $1 \ldots d$ and column $d+1 \ldots 2d$ and name the submatrix as W.

2. We will then prune $\mathcal{T}_{\text{large}}^{(t)}$ according to Lemma D.9 to output ϵ-approximation of $X_{:,i} \oplus \left(W^{-1}\mathcal{P}_\perp W_v^{(l)} X_{:,i}\right) \oplus \boldsymbol{A}_{:,i}$ for some vectors $\boldsymbol{A}_{:,i}$. As W is defined as the submatrix pruned by $W_V^{(t+1)}$, it holds that $W_V^{(t+1)'}\left(X_{:,i} \oplus \left(W^{-1}W_v^{(l)}X_{:,i}\right) \oplus \boldsymbol{A}_{:,i}\right) = \mathcal{P}_\perp W_v^{(l)} X_{:,i} \oplus 0^{m_{\text{large}}-m}$.

3. We will then prune $W_K^{(t+1)}$ and $W_Q^{(t+1)}$ according to Lemma D.7 to approximate attention patterns. We will choose boolean matrix M_K, M_Q such that for any $x \in \mathbb{R}^d$ and $a \in \mathbb{R}^{m_{\text{large}}-m}$,

$$\|(M_K \odot W_K^{(t+1)})^\top (M_Q \odot W_Q^{(t+1)}(x \oplus a)) - \left((W_K^{(l)})^\top W_Q^l x\right) \oplus 0^{m_{\text{large}}-m}\| \le \epsilon \|x\|_2.$$

   We can then have that the attention pattern for the large transformer at layer $t+1$ can approximate the small one. That is, for any $x \in \mathbb{R}^d$, $\|x\|_2 = O(1)$ and $a \in \mathbb{R}^{m_{\text{large}}-m}$,

$$\left\|\sigma\left((x \oplus a)^\top (M_K \odot W_K^{(t+1)})^\top (M_Q \odot W_Q^{(t+1)}(x \oplus a))\right) - \sigma\left(x^\top \left((W_K^{(l)})^\top W_Q^l x\right)\right)\right\| \le O(\epsilon).$$

   Combined with previous approximation on $W_V^{(t+1)'}\left(X_{:,i} \oplus \left(W^{-1}W_v^{(l)}X_{:,i}\right) \oplus \boldsymbol{A}_{:,i}\right)$ and the Lipschitzness of the LayerNorm, we have that the first m dimensions of the output after LayerNorm of the large Transformer at layer $t+1$ can ϵ-approximate the output after LayerNorm of the smaller Transformer at layer l.

4. We will finally prune the MLP in the projection function of $\mathcal{T}_{\text{large}}^{(t+1)}$ to approximate $\mathcal{P}_\perp f^{(l)}$ with $f^{(l)}$ being the MLP in the projection function of the projection function of $\mathcal{T}^{(l)}$.

The proof is then complete. □

E. Experiments

E.1. Training Details

For Figure 1, we train 2-layer standard GPT on $\text{Dyck}_{2,4}$ with sequence length no longer than 28. For (a), we train with hidden dimension and network width 200 and learning rate 3e-4. For (b), (c), (d), we train with hidden dimension and FFN width 50 and learning rate 3e-3.

For Figure 2, for (a), we train 1-layer transformer without residual link, FFN and the final LayerNorm before the linear head. The hidden dimensions and FFN widths are fixed as 500. For (a), we train the network with learning rate 1e-2 and for (b), (c), (d) we train the network with learning rate 3e-3.

let $\boldsymbol{o}_t$ denote the one-hot embedding where $\boldsymbol{o}_t[t] = 1$,

$$\boldsymbol{e}(\tau_{t,d}) = \boldsymbol{o}_{t \times D + d}, \tag{Type 1}$$

$$\boldsymbol{e}(\tau_{t,d}) = \boldsymbol{o}_t \oplus [\cos(\theta_d), \sin(\theta_d)], \tag{Type 2}$$

$$\theta_d = \arctan\left(d/(D+2-d)\right),$$

$$\boldsymbol{e}(\tau_{t,d}) = \boldsymbol{o}_t \oplus \boldsymbol{o}_d. \tag{Type 3}$$

Operator $\oplus$ means the concatenation of two vectors. Equation (Type 1) is the standard one-hot embedding for $\tau_{t,d}$. and Equation (Type 3) is the concatenation of one-hot embedding of types and depths. Finally, Equation (Type 2) is the embedding constructed in Yao et al. (2021).

E.2. Guiding The Transformer To Learn Balanced Attention

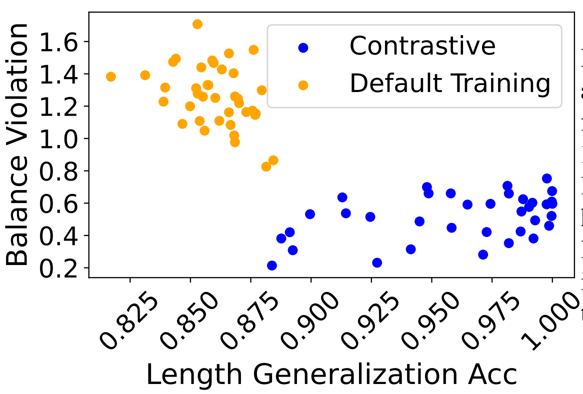


Figure 6: **Relationship Between Balance Violation and Length Generalization.** Accuracy from Transformers with minimal first layer with embedding Type 1, using both standard training and contrastive regularization (Equation (30)). Standard training leas to high balance violations which negatively correlate with length generalization performance. Contrastive regularization helps reduce the balance violation and improve the length generalization performance.

In our experiments, we observe that although models learned via standard training that can generalize well in distribution, the length generalization performance is far from optimal. This implies that the models are not finding the correct algorithm for parsing Dyck when learning from finite samples. A natural question is: can we guide Transformers towards correct algorithms, as measured by better generalization on longer Dyck sequences?

In the following, we measure length generalization performance by testing the accuracy of the model on valid Dyck prefixes with length randomly sampled from 400 to 500, which approximately correspond o 16 times the length of the training sequences. We will show generalization can be improved by regularizing the attentions to be more balanced, inspired by results in Section 3.

Balance violation negatively correlates with length generalization accuracy We denote the *balance violation* of a Transformer as $\beta := \mathbb{E}_{d,d',i,j}\left[S_{d,d',i,j}/P_{d,j}\right]$ for S, P defined in Equations (5) and (7). Theorem 3.2 predicts that for models with a minimal first layer, perfect length generalization requires β to be zero. Beyond such idealized condition, it is natural to ask whether a small yet positive β correlates with length generalization accuracy in practice. Our results show a moderate correlation (-0.38 `SpearmanR` with p-value 0.014) based on over 40 random initializations (Figure 6).

Given the correlation, we design a contrastive training objective to reduce the balance violation, which ideally would lead to improved length generalization. Specifically, let p_r denote a prefix of r nested pairs of brackets of for $r \sim U([D])$, and let $\mathcal{T}(s \mid p_r \oplus s)$ denote the logits for s when $\mathcal{T}$ takes as input the concatenation of p_r and s. We define the *contrastive regularization* $R_{\text{contrastive}}(s)$ as the mean squared error between the logits of $\mathcal{T}(s)$ and $\mathcal{T}(s \mid p_r \oplus s)$, taking expectation over r and p_r:

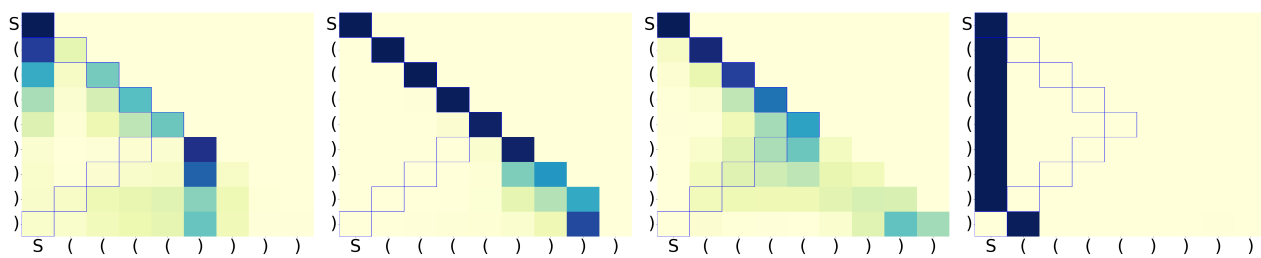
$$\mathbb{E}_{r\sim U([D]),p_r}\left[\|\mathcal{T}(s \mid p_r \oplus s) - \mathcal{T}(s)\|_F^2\right]. \tag{30}$$

Following the same intuition as in the proof of Theorem 3.2, if the model can perfectly length-generalize, then the contrastive loss will be zero. We then train the model with contrastive loss and observe that the balance violation is reduced and the length generalization performance is improved (Figure 6).

E.3. Additional Results on Dyck Prefix

In the experiment presented in the main text, we perform experiments on complete Dyck sequences, which is a special case of Dyck prefixes. In this section, we present additional experiments on Dyck prefixes $\mathsf{Dyck}_{2,4,28}$.

Attention Patterns We first perform experiments on attention patterns. The qualitative results are shown in Figures 7 and 9. We can observe that the attention patterns are still diverse and do not commonly show stack-like patterns. We also calculate the *attention variation* [9], and find that the attention variation is 0.34, based on 30 models with a minimal first layer and different random seeds. In contrast, for models with a standard first layer and without position encodings, the attention variation is surprisingly high, reaching 14.51. The high value is caused by the large distance between attention patterns like Figure 7 (c) and (d); that is, between patterns that attend more to the current positions, and patterns that attend more heavily to the initial position. The difference is even increased when we consider longer sequence (Figure 8). Similarly, the variation is also high for models with linear position embedding, reaching 11.92. This shows that the attention patterns are still diverse and do not commonly show stack-like patterns.

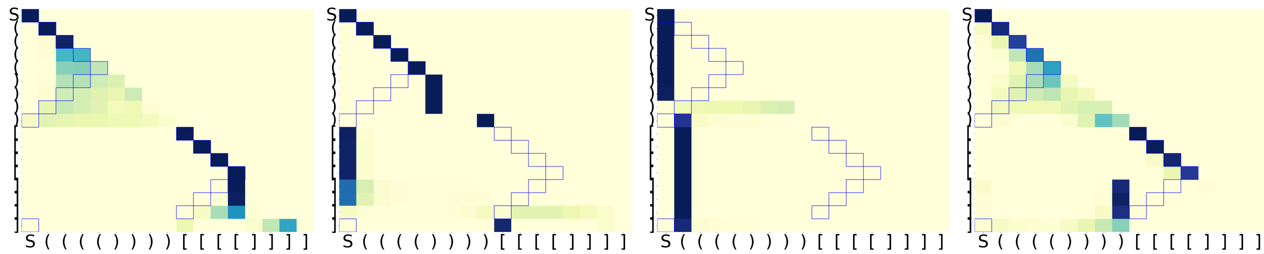


(a) With Position Embedding (b) With Position Embedding (c) Without Position Embedding (d) Without Position Embedding

Figure 7: **Second-layer attention patterns of two-layer Transformers on Dyck Prefix**: Models for (a),(b) are under the same setup but different random seeds; similarly for (c),(d). All models reach $\geq 97\%$ accuracy (defined in **??**). In the heatmap, darker color indicates larger value. As we can observe, the attention patterns still show much variance.

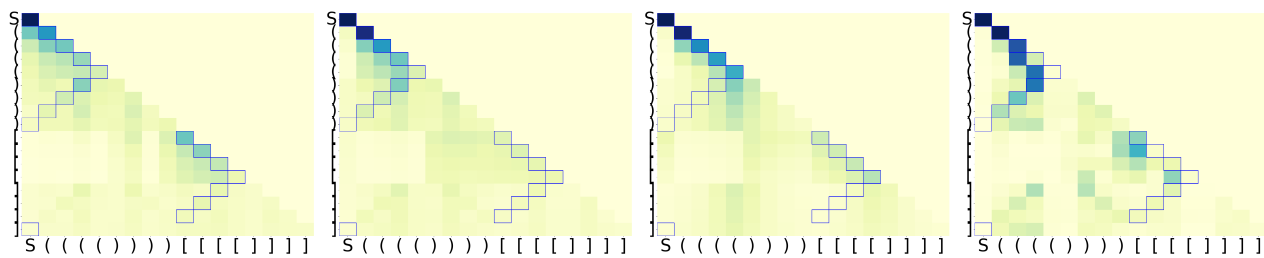
Balanced Violations We also test the relationship with the balance violation with length generalization on Dyck prefixes, similar to Figure 6. We observe that although the negative correlation is not presented as in the case of Dyck sequences, contrastive regularization still helps reduce the balance violation and significantly improve the length generalization performance. This shows that for Dyck prefixes, while the balance violation may not be predictive of the length generalization performance, it is still possible to reduce the balance violation and improve the length generalization performance. The results are shown in Figure 10.

[9] Recall from **??** that the attention variation between two attention patterns $A_1, A_2 \in \mathbb{R}^{N\times N}$ is defined as $\text{Variation}(A_1, A_2) = \|A_1 - A_2\|_F^2$.



(a) With Position Embedding Run 1 (b) With Position Embedding Run 2 (c) Without Position Embedding Run 1 (d) Without Position Embedding Run 2

Figure 8: **Second-layer attention patterns of two-layer Transformers on Longer Dyck Prefix**: Models for (a),(b) are under the same setup but different random seeds. All models reach $\geq 97\%$ accuracy (defined in **??**). In the heatmap, darker color indicates larger value.



(a) Embedding Type 1, run 1 (b) Embedding Type 1, run 2 (c) Embedding Type 2 (d) Embedding Type 3

Figure 9: **Second-layer attention patterns of two-layer Transformers with a minimal first layer**: (a), (b) are based on embedding Type 1 with different random seeds. (c), (d) are based on embedding Type 2 and Type 3. Different embedding functions lead to diverse attention patterns, most of which are not stack-like.

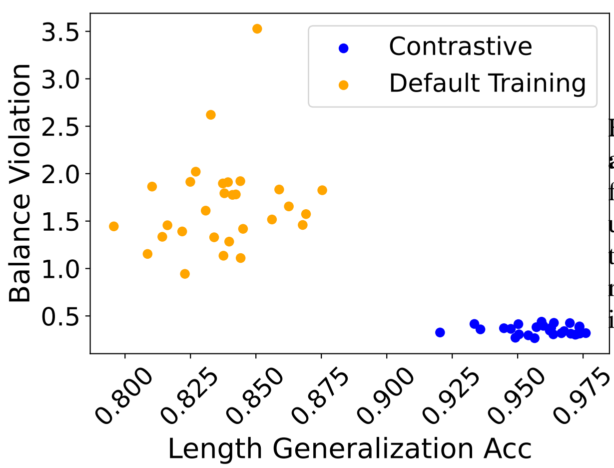


Figure 10: **Relationship Between Balance Violation and Length Generalization.** Accuracy from Transformers with minimal first layer with embedding Type 1, using both standard training and contrastive regularization (Equation (30)). We again observe that contrastive regularization helps reduce the balance violation and improve the length generalization performance.